UNIVERSITAT DE VALÈNCIA
DEPARTAMENT D'INFORMÀTICA
I ELECTRÒNICA

Algoritmos de soporte para Tolerancia a Fallos. Aplicación a la Comunicación Punto a Punto

TESIS DOCTORAL

Presentada por:
D. Vicente Cerverón Lleó
Dirigida por:
D. Gregorio Martín Quetglás

Valencia, 1996

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Física

UNIVERSITAT DE VALÈNCIA

UNIVERSITAT DE VALÈNCIA

DEPARTAMENT D'INFORMÀTICA
I ELECTRÒNICA

C/ Dr. Moliner, 50
46100 Burjassot (VALÈNCIA)

D. Gregorio Martín Quetglás, Catedrático de Ciencias de la Computación e Inteligencia Artificial del Departament d'Informàtica i Electrònica de la Universitat de València,

HACE CONSTAR que el licenciado en Ciencias Físicas D. Vicente Cerverón Lleó ha realizado bajo su dirección y en los laboratorios del Departamento de Informática y Electrónica, el trabajo titulado **"Algoritmos de soporte para Tolerancia a Fallos. Aplicación a la Comunicación Punto a Punto"**, que se presenta para optar al grado de Doctor en Ingeniería Informática.

Y para que así conste, firmo el presente certificado en Valencia a treinta de diciembre de mil novecientos noventa y seis.

Fdo: Gregorio Martín

Agradecimientos

Muchas personas han influido en mi camino dentro de la universidad durante el periodo transcurrido desde que obtuve el título de licenciado hasta llegar a optar al grado de doctor; afortunadamente, la mayoría de ellas lo han hecho para bien. Espero no dejarme a ninguna de ellas en estos agradecimientos, y si alguna se me escapa le ruego sea tolerante a mis fallos.

En primer lugar quiero manifestar mi agradecimiento al director de este trabajo, Gregorio Martín, quien con su asesoramiento científico, sus consejos en la dirección, su estima y el empuje que siempre me ha transmitido ha hecho posible esta memoria de investigación, al igual que ha hecho posible otras muchos proyectos aún más complejos dentro de esta universidad.

Igualmente quiero expresar mi agradecimiento a Joan Pelechano y Marcelino Vicens, quienes me permitieron entrar en este Departamento de Informática y Electrónica, me iniciaron en el mundo de la investigación y la docencia y me han animado constantemente en mi camino hasta aquí. También quiero citar aquí a José Espí, Director del Departamento durante casi todo este tiempo, quien siempre me ha mostrado su apoyo y ha depositado su confianza en mí.

No puedo dejar de nombrar a los compañeros de la Unidad Docente de Informática que han estado en el departamento desde un principio, con los cuales he tenido unas excelentes relaciones, hemos compartido diversos proyectos y me han apoyado en todos mis pasos; especialmente Juan de Mata, Jesús, Vicente y Paco; también todos los demás. Quiero además agradecer el trato recibido por el resto de compañeros del departamento (secretaría incluida) y por una larga lista de personas dentro de la comunidad universitaria, con un especial agradecimiento a Carlos.

Un párrafo adicional merecen todos los compañeros del LISITT empezando por mis compañeros en el proyecto FASST, Rafa, Carlos, Fernando, José Antonio y Germán. Como saben todos los que conocen el LISITT la lista es casi interminable, a parte de variable (incrementalmente) por lo cual no los nombraré personalmente; daos todos por agradecidos al leer esto. Quedo al servicio de todos vosotros y animo a todos los que tienen que defender sus tesis en un futuro no lejano. Reflejo aquí la cordialidad recibida, con un cariño especial para Ariadna y para los también nómadas, que tanto me han ayudado y animado, Ricardo, Vicente y Elena, quien ha deseado el buen fin de este trabajo casi tanto como el del suyo propio.

Por último, quiero agradecer sinceramente a Julián y a Vicente Cavero (de nuevo) la ayuda imprescindible e impagable que me han prestado en la experimentación de este trabajo y en la revisión de la memoria.

A mis padres, por su cariño y dedicación y su voluntad de que alcance metas mayores.

A mis hermanos, familia y amigos, que desean siempre lo mejor para mí.

A Paula, Juan y Javier, que tantas alegrías me dan y
a los que a veces no dedico el cariño y la paciencia que merecen.

A María José,
por y gracias a ella he terminado este trabajo,
con todo mi amor.

ÍNDICE GENERAL

INTRODUCCIÓN

La tolerancia a fallos es un paradigma acuñado históricamente desde el área de la Arquitectura de Computadores donde siempre han sentido la preocupación de que los ordenadores pudieran trabajar con unos niveles de seguridad cada vez mayores, de tal forma que se pudiera confiar en la máquina, en las situaciones más cruciales del control de cualquier proceso. Los sistemas redundantes y triple redundantes fueron las primeras aproximaciones al problema y seguramente su desarrollo hubiera sido mucho mayor por un lado si la industria del software hubiera podido dar las mismas prestaciones en materia de tolerancia a fallos, y por otro si los desarrollos en materia de comunicaciones que se han dado en los últimos años hubieran coincidido con la evolución de los resultados que los diseñadores de computadores ponían sobre la mesa.

Hay que indicar que en este campo de la tolerancia a fallos, Europa ha sufrido algunos retrocesos importantes; la apuesta anglo-francesa por un procesador completamente seguro (recuérdese la historia de VIPER) constituyó un completo fracaso, mientras las soluciones por la vía de la arquitectura y del software (surgidas principalmente en USA) acabarían dando resultados satisfactorios, imponiendo una situación que ahora nos parece obvia: la solución estriba en "conectar y duplicar adecuadamente elementos", no en buscar "elementos 100% seguros", curiosamente la misma solución que toma la naturaleza a la hora de transmitir el material genético, donde la respuesta está en la redundancia, la velocidad de proceso y la interrelación [Mar96].

0.1. ANTECEDENTES

Los antecedentes de la investigación que ha dado lugar a esta memoria se situan en el inicio en 1990, con la participación de la Universitat de València en el proyecto P5212 "*Fault-tolerant Architecture for Stable Storage Technology*" (FASST) [FASST90]

financiado por la entonces CEE dentro del programa ESPRIT-II que perseguía establecer una arquitectura para computadores tolerantes a fallos a todos los niveles posibles [Fab96][Cog97]. En este proyecto, una aplicación de tráfico que entonces se estaba instalando, los carriles reversibles de la Diagonal de Barcelona, fue considerada como una buena candidata para comprobar las bondades de una nueva arquitectura tolerante a fallos. Como muchos otros proyectos europeos, el proyecto FASST no consiguió llegar al final de su Anexo Técnico, y por el camino del proyecto quedaron compañías como la británica Ferranti (cuyo final tuvo mucho que ver con el enorme fiasco del citado VIPER), la interminable agonía de Bull que tampoco consiguió situarse en el mercado de la tolerancia a fallos y la desaparición de la alemana Stollmann (a la sazón contratista principal del proyecto), pruebas de cierta incapacidad empresarial europea para hacer frente a las nuevas tecnologías informáticas [Pow94][BAH97].

A pesar de estas dificultades, en el LISITT (Laboratorio Integrado se Sistemas Inteligentes y Tecnologías de la Información en Tráfico y Transporte) quedaron tres líneas que están dando sus resultados en el campo académico:

- los sistemas operativos tolerantes a fallos [Per97]

- la finalización de una placa tolerante a fallos y la inyección de errores [RMD97]

- el análisis de la funcionalidad y necesidades de sistemas tolerantes a fallos en el marco de la instrumentación de la Gestión de Tráfico, línea que se ha continuado hasta la finalización de la presente memoria.

En lo que llevamos de década, ha ido tomando cuerpo una nueva subrama de la ingeniería que los europeos empezamos llamando RTI (*Road Transport Informatics*), luego denominamos ATT (*Advanced Telematics in Transport*) y que acabaremos designando como han dado en hacer los estadounidenses ITS (*Intelligent Transport Systems*), quienes habiendo llegado más tarde al tema parece que acabarán por organizarlo definitivamente.

Hemos de reconocer que sin este trabajo sistemático llevado a cabo al otro lado del Atlántico, esta memoria posiblemente no existiera en su actual formato, aunque sí sus resultados más significativos, ya que el esfuerzo llevado a cabo en el proyecto FASST puso de manifiesto una serie de requisitos específicos de la Telemática en Tráfico respecto a la Tolerancia a Fallos. La aparición durante los últimos tres años de este trabajo sistemático sobre las arquitecturas de ITS, ha ayudado a ubicar nuestra línea de trabajo centrada en la búsqueda de sistemas donde tanto los nodos como sus conexiones fueran tolerantes a fallos.

La cuestión básica que ha animado nuestro trabajo ha sido analizar las posibilidades que podían existir de aplicar técnicas propias de la arquitectura de ordenadores tolerantes a fallos a aquellas comunicaciones dentro de los sistemas de ITS que presentan unos especiales requisitos de fiabilidad todavía no resueltos plenamente. Aun asumiendo que los actuales sistemas electrónicos de comunicación han resuelto casi todos los aspectos de velocidad y exactitud en la transmisión de la información, en nuestro campo hablamos de equipos y líneas que están en instalaciones de campo abierto, donde muchos errores insospechados en ambientes cerrados se presentan con mucha más frecuencia de la que podría sospecharse "a priori".

0.2. PLANTEAMIENTO

El transporte por carretera de personas y bienes se ha convertido en una actividad estratégica en la relación entre ciudades y regiones, y un hecho físico que afecta cada vez a más personas y durante más tiempo, constituyéndose en una preocupación social de la mayor transcendencia debido al incremento de la población que vive en áreas metropolitanas, con el consiguiente crecimiento de los viajes dentro de y hacia ellas en zonas donde la capacidad viaria actual es insuficiente, todo ello tanto para el transporte de personas como para el de mercancías.

Conforme se ha ido desarrollando estas necesidades de desplazamiento e interrelación, el sector del transporte debe enfrentar serios impedimentos que pueden frustrar la potencialidad de la economía global. Algunos ejemplos de estos inconvenientes son:

los costes sociales en vidas humanas de la inseguridad del tráfico,

la contaminación y el deterioro ambiental y

la congestión del tráfico y su impacto socioeconómico.

Para hacer frente a estos problemas, se han desarrollado una serie de técnicas que contribuyen a incrementar la eficiencia del tráfico en su conjunto y a reducir sus efectos negativos [Fer96]. La respuesta al conjunto de los problemas del transporte no es única. Precisamente en las zonas con mayores problemas de tráfico, no hay opción para construir más carreteras ni aumentar el tamaño de las existentes debido a la falta de terreno adecuado, a la limitación de los recursos financieros y al impacto que ello supondría sobre el medio ambiente. Uno de los avances tecnológicos que pueden mejorar la eficiencia de los sistemas de transporte es la introducción de nuevas tecnologías de la información, de la sensorización y de las telecomunicaciones en lo que se ha dado en llamar sistemas de

transporte inteligentes (ITS. [1]), que pretenden mejorar la movilidad sin necesidad de aumentar las infraestructuras viarias.

Algunas de estos sistemas son métodos nuevos y mejores para realizar actividades tradicionales, como el control del tráfico. Otros sistemas son nuevos por completo, como los sistemas de navegación y guiado dinámico. La mayoría son ideas que los profesionales del transporte sostienen desde hace tiempo pero que hasta el momento estaban fuera del alcance de la tecnología disponible o de la rentabilidad económica (aun hoy, determinadas aplicaciones no son rentables por separado, y sólo llegan a serlo cuando se integran con otras aplicaciones del transporte o de otras áreas). La aplicación de las tecnologías de ITS constituye una colección de servicios al usuario, agrupados en siete áreas funcionales [NPB96]: sistemas de información, gestión del transporte público, gestión de flotas, gestión del tráfico, gestión de la demanda, gestión de aparcamientos y asistencia a la conducción.

El abanico de tecnologías y opciones disponibles para intentar resolver los problemas asociados al tráfico y al transporte permite a los desarrolladores una variedad de opciones para cubrir sus necesidades. Sin una coordinación adecuada, se corre el riesgo de desarrollar sistemas que cubren sus necesidades pero que son incompatibles con sistemas que deben basar sus servicios en una misma infraestructura telemática, y por consiguiente utilizando soluciones estándares, o son incompatibles con sistemas que proporcionan un mismo servicio en áreas geográficas vecinas (la llamada continuidad del servicio) [NAR96s]. Dicho de otro modo, si la ciudad A elige implementar un servicio de un modo, y la ciudad B decide implementarlo de otro modo diferente, existe la posibilidad de que el equipamiento o servicios que un conductor contrate para uso en la ciudad A sean inservibles en la ciudad B. Otro tanto sucede con los servicios a usuarios institucionales: dos ciudades vecinas que implementan sistemas de gestión y control de tráfico diferentes son incapaces de compartir datos y mantener un funcionamiento coordinado.

---

[1]En lo sucesivo se empleará en esta memoria el acrónimo ITS proveniente de sus iniciales en inglés Intelligent Transport Systems. Aunque tanto el doctorando como su director coinciden en que no es deseable la importación de términos extranjeros, la rápida evolución de las tecnologías y la difusión en la comunidad científica y técnica internacional de determinados acrónimos de uso frecuente aconseja, en la mayoría de los casos en que se empleen abreviaturas, la utilización de los acrónimos anglosajones con preferencia a la creación de nuevos acrónimos fruto de las iniciales castellanas de cada denominación. En cualquier caso, cada vez que se introduzca un acrónimo se indicará su significado y procedencia.

Diversos grupos de trabajo, desde el inicio de los 90, tratan de establecer un modelo de referencia para la arquitectura de los ITS así como una terminología común. Por un lado el Comité Europeo para la Normalización (CEN) creó en 1991 un Comité Técnico TC 278 "Telemática del Tráfico y Transporte por Carretera". Uno de sus grupos de trabajo (WG13) "Arquitectura y terminología" está dedicado a esta tarea, en supuesta colaboración con la labor del grupo de trabajo WG1 "Arquitectura" del Comité Técnico TC 204 "Sistemas de control e información de tráfico" de la ISO (Organización Internacional para la Estandarización), ambos grupos coordinados bajo la supervisión de éste último [Pat95]. La documentación más valiosa y abundante acerca de este tipo de trabajo es la procedente del Departamento de Tráfico de los Estados Unidos de América [NAR96] que, como tendremos ocasión de revisar, ha establecido una arquitectura para los ITS consistente en una arquitectura lógica de funciones, una arquitectura física de elementos y una matriz de interrelación, que muestra los flujos de información entre entidades físicas [ITS96] (véase en el anexo B de esta memoria la revisión europea acerca de la arquitectura americana).

Las necesidades de armonización son especial y obviamente relevantes en la comunicación de datos en los ITS. Las aplicaciones relacionadas con la gestión del tráfico y del transporte han empleado tradicionalmente la comunicación de datos para la transmisión hacia un centro de control de tráfico de los datos obtenidos a través de sensores situados en la vía, y para la transmisión de órdenes y mensajes desde el centro de control hacia los dispositivos de información y/o señalización [Obi96]. Sin embargo, estas aplicaciones se han planteado tradicionalmente sobre el uso de comunicaciones utilizando una red privada específica instalada para servir a la aplicación intercambiándose los datos mediante protocolos desarrollados para cada aplicación. Las revisiones existentes demuestran que estos sistemas en su conjunto se han venido diseñando como un universo cerrado, en principio no preparado para interaccionar con otros sistemas de gestión o información [RKR96].

Aunque como hemos indicado anteriormente los sistemas ITS y sus precursores han venido conociéndose también como ATT (*Advanced Transport Telematics*), hay que hacer notar que sólo se puede hablar con propiedad de telemática aplicada al transporte cuando las comunicaciones empleadas en las aplicaciones relacionadas con el transporte utilizan los medios y herramientas (tanto físicos como lógicos) comúnmente extendidos en el mundo de la tecnología de las comunicaciones. El uso de estos elementos, particularmente en lo que respecta a los protocolos de comunicaciones generalizados en las redes telemáticas, permite la integración de las redes locales dedicadas a la transmisión de datos del tráfico y el transporte dentro de las redes telemáticas de uso general, expandiendo de este modo las posibilidades de interacción entre sistemas.

El desarrollo de los ITS requiere la adopción de estándares vigentes en el campo de las telecomunicaciones de propósito general y sólo en casos muy particulares se justifica el desarrollo de nuevas propuestas para las comunicaciones específicas de los ITS y sus aplicaciones. Un ejemplo sobresaliente de ello es el NTCIP (*National Transportation Communications for ITS Protocol*) un conjunto de estándares para la comunicación entre los centros de control de tráfico y los dispositivos situados en la carretera [NTCIP96f] que pretende emplear los estándares existentes más adecuados para los ITS, basándose en el modelo de referencia OSI (*Open System Interconnection*), al objeto de permitir la interoperatividad entre distintos dispositivos de control de tráfico empleando una única infraestructura de comunicaciones de uso común a todos ellos. El conjunto de protocolos propuestos se basa en la pila de protocolos Internet en lugar del X25 seguido por un gran número de las redes de comunicaciones existentes en sistemas de tráfico [Sny95].

Los sistemas de información y comunicaciones usados en ITS están sujetos a fallos y averías de resultados diversos según la aplicación que se trate y el alcance de la incorrección, pudiendo ir desde una simple molestia para los usuarios hasta resultados catastróficos en un sistema de control crítico [Abb90]. Por ello a la hora de diseñar y construir nuevos sistemas informáticos deben tenerse en cuenta, a parte de la potencia de cálculo, otros factores; estos factores adicionales a considerar son tres: **fiabilidad, disponibilidad y seguridad** [Lap85].

Entenderemos por fiabilidad la propiedad de un sistema que permite confiar de manera justificada en los servicios que el sistema debe proporcionar. La disponibilidad de un sistema indica la fracción del tiempo durante el cual dicho sistema cumple con los servicios esperados. Por su parte, la seguridad de un sistema trata de determinar la disposición del mismo para evitar consecuencias catastróficas de un fallo sobre el entorno [Lap92].

Un sistema ideal sería aquel completamente fiable, de modo que no presentase ningún fallo; este sistema estaría siempre disponible para su uso y sus resultados serían 100% seguros. La experiencia muestra que no existen tales sistemas aunque resulta importante tratar de aproximarse a este tipo de requisitos. La metodología de la tolerancia a fallos se basa actualmente en la consideración de que todo sistema digital es susceptible de fallar de distintas maneras. Por tanto deben identificarse los fallos posibles que pueden aparecer, y de estos cuáles son los más habituales.

En base a la catalogación de los posibles fallos, se busca diseñar y construir sistemas tolerantes a los mismos. La técnica básica a emplear, en sus distintas formas, es la **redundancia en el diseño** [Avi75]. Aunque tendremos ocasión de tratar los tipos de

redundancia con más profundidad, a nivel introductorio señalemos los tres más importantes: **redundancia espacial**, que refiere la existencia de más recursos de los estrictamente necesarios para el funcionamiento del sistema [Hop78]; **redundancia temporal**, que indica la posible redundancia de una acción en el dominio del tiempo [Ran75][Swe91]; y la **redundancia de la información**, que expresa la presencia de contenidos adicionales para mantener la integridad de la información [Nel87][Swe91]. La redundancia busca permitir el **funcionamiento continuo** del sistema incluso en presencia de un fallo o, en caso de que eso no sea posible, se pretende minimizar el daño causado, intentando diagnosticar y reparar el sistema en el menor tiempo posible. Todo ello dependerá de la naturaleza de la aplicación y de las posibles consecuencias que acarrearía un funcionamiento incorrecto según la duración del mismo.

Los métodos de tolerancia a fallos inicialmente se emplearon en unas pocas aplicaciones especializadas, como la navegación espacial [Hop78]. El coste asociado a la redundancia de hardware o software venía justificado por los daños económicos que causaría un error del sistema digital de control. En los últimos años, con el descenso del precio de los distintos elementos, las técnicas de tolerancia a fallos se han venido aplicando a un amplio rango de sistemas (control industrial, control de materiales peligrosos, transacciones bancarias y comerciales, etc.) aplicando distintas técnicas en función de los diferentes objetivos marcados por la aplicación.

Los Sistemas de Control de Tráfico han sido también objeto de esfuerzos para incrementar la fiabilidad al ser considerados ciertos subsistemas como críticos, por el impacto de los posibles fallos, especialmente en cuanto a seguridad vial. Los sistemas y dispositivos de última generación incluyen mecanismos a prueba de fallo para garantizar su correcto funcionamiento [Boy96]. En algunas ocasiones la tolerancia a fallos se analiza de manera incorrecta. Por ejemplo el departamento de tráfico de una gran ciudad española adquirió un gran computador tolerante a fallos (Stratus) para el Centro de Control de Tráfico, planteamiento que hay que considerar como exagerado, ya que el diseño tolerante a fallos se basa en un compromiso entre la fiabilidad y el coste [Lee90]. Parece más lógico destinar recursos adicionales para tolerar aquellos fallos más frecuentes especialmente en los equipos y subsistemas críticos. Los fallos más frecuentes en los sistemas de control de tráfico se producen en los enlaces de comunicación y en los dispositivos de campo (sensores y sistemas de señalización), mientras que los fallos más críticos son los de los sistemas de señalización por su efecto sobre la seguridad vial. Nótese que los sistemas de señalización de tráfico fueron los primeros en incorporar mecanismos a prueba de fallos;

valga como ejemplo la gestión de los semáforos de una intersección, que en caso de fallo generalizado están preparados para mantener la señalización ámbar.

La comunicación de datos es reconocida como fuente potencial de problemas, en tanto en cuanto los datos pueden ser alterados o perdidos fallando la transmisión por muy diversas causas externas. Se ha desarrollado pues en este área un trabajo considerable para conseguir una comunicación fiable, libre de errores [BJ87][Swe91]. Sin embargo, los algoritmos empleados para buscar una comunicación fiable no pueden hacerlo a cualquier precio: la infraestructura necesaria y la eficiencia de la transferencia de datos es un factor a tener en cuenta conjuntamente a la fiabilidad de la misma.

0.3. OBJETIVOS DE LA INVESTIGACIÓN

Teniendo en cuenta las consideraciones anteriores y la situación de la bibliografía, nos pareció pertinente plantear los siguientes temas de estudio:

1. **Determinar** si existen aplicaciones críticas en los ITS con requisitos especiales de fiabilidad no completamente resueltos, desde la metodología de tolerancia a fallos. Para ello necesitamos analizar los sistemas de los ITS existentes, el estado actual en el desarrollo y estructuración de los mismos, los esfuerzos de estandarización de las comunicaciones en ITS y la relación de estos estándares con los estándares adoptados en las comunicaciones de propósito general.

2. En el caso de que el análisis del punto anterior identificara algún tipo de carencia, **profundizar** en los problemas que presentan las soluciones existentes de tolerancia a fallos, especialmente en lo referente a la tolerancia a fallos en las comunicaciones, para su integración dentro de la arquitectura de ITS al objeto de proponer métodos estandarizables para incrementar la fiabilidad de las comunicaciones críticas en los ITS que sean compatibles con los estándares de la tecnología de comunicaciones en general y que superen los problemas de integración con las propuestas de normalización para comunicaciones en los ITS.

3. **Extraer** una ontología de sistemas y, asumiendo que los enlaces punto a punto constituyen una parte vertebral de la arquitectura ITS, **desarrollar** en consecuencia algoritmos para tolerancia a fallos con una visión global de los mismos aplicable al análisis y definición de métodos de gestión de redundancia sobre estos enlaces, basándose en el protocolo PPP de Internet y sus extensiones. Este objetivo viene indicado por la relevancia del PPP en la tecnología de las comunicaciones en general y su repercusión en los ITS en particular.

0.4. ESTRUCTURA DE LA MEMORIA

La memoria de esta investigación empieza por una exposición que trata fundamentalmente de establecer los conceptos y términos básicos utilizados para expresar todos aquellos aspectos relacionados con la fiabilidad y la tolerancia a fallos, comentando los desarrollos más extendidos en este campo y destacando el carácter general de éste modo de diseñar los sistemas, independientemente de que éstos sean físicos o lógicos. Para ello se establece una **metodología** general con un catálogo de modelos de operación para la utilización de la redundancia.

A continuación se inicia un bloque de dos capítulos que analiza una problemática de gran impacto: los sistemas de transporte inteligentes (ITS) y las tecnologías de las comunicaciones que los sostienen. En el primero de los capítulos de este bloque se expone una visión general de esta tecnología de los ITS para después incidir en las redes de comunicación utilizadas que son comúnmente menos estudiadas que las aplicaciones que deben funcionar sobre dichas redes. Se revisan los trabajos tendentes a normalizar las redes de comunicaciones sobre las que basar los ITS. Esta revisión se complementa con el anexo B que recoge una revisión de expertos europeos sobre la arquitectura estadounidense.

En el tercer capítulo se tratan a partir del capítulo anterior los requisitos especiales de fiabilidad de los sistemas de control de tráfico y de las comunicaciones empleadas en los mismos para: **analizar** las necesidades de tolerancia a fallos, **discutir** los problemas de integración de soluciones existentes dentro de la arquitectura y estándares existentes en los ITS y **proponer métodos** estandarizables de incrementar la fiabilidad de las comunicaciones críticas mediante diferentes formas de redundancia que sean integrables con las propuestas de normalización para comunicaciones en los ITS. Con este capítulo determinamos las aplicaciones de ITS críticas desde el punto de vista de fiabilidad y describimos los problemas encontrados en las soluciones actualmente existentes, cubriendo los .objetivos 1 y 2 que nos habíamos planteado.

Una vez establecidas las necesidades de nuevos desarrollos para la comunicación punto a punto en ITS, el cuarto capítulo empieza abordando el objetivo número 3. En él se describe el protocolo punto a punto PPP y sus extensiones, con una exposición de las características relevantes para el problema que se trata en esta memoria.

El quinto capítulo presenta la aplicación de los modelos del capítulo primero a la comunicación punto a punto, describiendo un servicio de enlace de datos con enlaces PPP simples y gestión de la redundancia mediante el protocolo MLPPP. Utilizando la ontología

extraída, desarrollamos procedimientos de gestión de la redundancia y actividades de tolerancia a fallos correspondientes a los modelos establecidos en el capítulo primero, con un análisis de las ventajas e inconvenientes de cada uno de ellos que produce criterios de selección para desarrollos posteriores.

El capítulo sexto lo constituye la aplicación de los análisis y de las propuestas seleccionadas al estudio de un caso de un sistema ITS relacionado con el control lineal de la velocidad donde se manifiesta: la factibilidad de emplear enlaces punto a punto, las ventajas de emplear un protocolo de enlace de datos estandarizado y difundido como es el PPP de Internet, y la conveniencia de instalar enlaces redundantes y gestionarlos mediante el protocolo Multienlace PPP que permite mantener la interoperatividad con los protocolos estandarizados de los niveles superiores. Ello supone el desarrollo de una arquitectura para el sistema que soporta el servicio en cuestión siendo simultáneamente tolerante a fallos y compatible con la actual arquitectura global de ITS.

El séptimo capítulo (complementado con el código presentado en el anexo A) expone la implementación y prueba de los algoritmos que se han considerado más adecuados para las aplicaciones tratadas. En el describimos la codificación, inclusión y comprobación desarrollados en un sistema operativo Linux, seleccionado por su carácter público. Sobre esta plataforma, partiendo de una versión existente que soporta el protocolo PPP, se introducen las variaciones y extensiones propuestas, especialmente la implementación de Multienlace PPP sobre enlaces PPP simples, y se expone el proceso de pruebas llevado a cabo y los resultados experimentales.

Los resultados obtenidos nos indican que la aplicación de la metodología general de tolerancia a fallos permite trasladar los diferentes modelos de gestión de la redundancia al desarrollo de algoritmos para la gestión de la redundancia de enlaces de comunicación punto a punto, todos ellos emplazados en un nivel de enlace de datos e implementables por software.

CAPÍTULO PRIMERO

CONCEPTOS DE FIABILIDAD Y MODELOS PARA TOLERANCIA A FALLOS EN SISTEMAS DE CONTROL

Construir un sistema absolutamente perfecto, incluso aunque pudiera garantizarse este objetivo, resulta más difícil y más costoso que construir un sistema suficientemente correcto provisto de cualidades de tolerancia a fallos. Cualquier sistema, independientemente del esfuerzo dedicado a su diseño y la calidad del proceso de producción, una vez en funcionamiento, se verá expuesto a efectos o perturbaciones externas, así como al progresivo desgaste de sus componentes. Ambos factores pueden hacerlo desviarse de su comportamiento esperado y convertir el mejor de los sistemas en un equipo inoperante, si no perjudicial.

1.1. CONCEPTOS BÁSICOS EN SISTEMAS FIABLES

Muchos términos pueden ser usados para describir el deseo de que un sistema cumpla sus objetivos sin fallar. La mayoría de estos términos tienen dos características: primero, que el usuario del sistema (persona u otro sistema) precisa algún tipo de servicio del sistema; y segundo, el usuario depende del (confía en el) cumplimiento de ese servicio. Bajo estas consideraciones, surge la siguiente definición [Lap85][Lap92]

Garantía de funcionamiento *(dependability)*:
es la propiedad de un sistema (computacional) que permite confiar de manera justificada en los servicios que el sistema debe proporcionar.

Esta definición general presenta múltiples facetas, según los tipos de aplicaciones, Lógicamente, aplicaciones diferentes requerirán diferentes garantías de funcionamiento, o se concentrarán en aspectos diferentes de la misma y en consecuencia usarán otros

términos específicos en lugar de éste. Gran parte del trabajo en tolerancia a fallos tiene por objetivo mejorar la **fiabilidad** *(reliability)* de un sistema, como capacidad del sistema para cumplir normalmente sus servicios. Algunas aplicaciones precisan aspectos particulares de la garantía de funcionamiento como son la necesidad de máxima **disponibilidad** *(availability)*, considerada como porcentaje de tiempo durante el cual el sistema cumple con los servicios esperados.

En aplicaciones críticas, el énfasis se pone en evitar consecuencias catastróficas del fallo en el entorno [Abb90]; este enfoque lleva a consideraciones sobre la **seguridad** *(safety)* de un sistema. En otras áreas, el énfasis se sitúa mas bien en términos de la privacidad de la información y la autorización de accesos; esto se conoce como **inviolabilidad** *(security)* de un sistema.

Dado que el funcionamiento incorrecto del sistema tiene diferentes consecuencias según las aplicaciones, el término "garantía de funcionamiento" parece apropiado de manera genérica para expresar la necesidad de que un sistema cumpla con su función [Lee90].

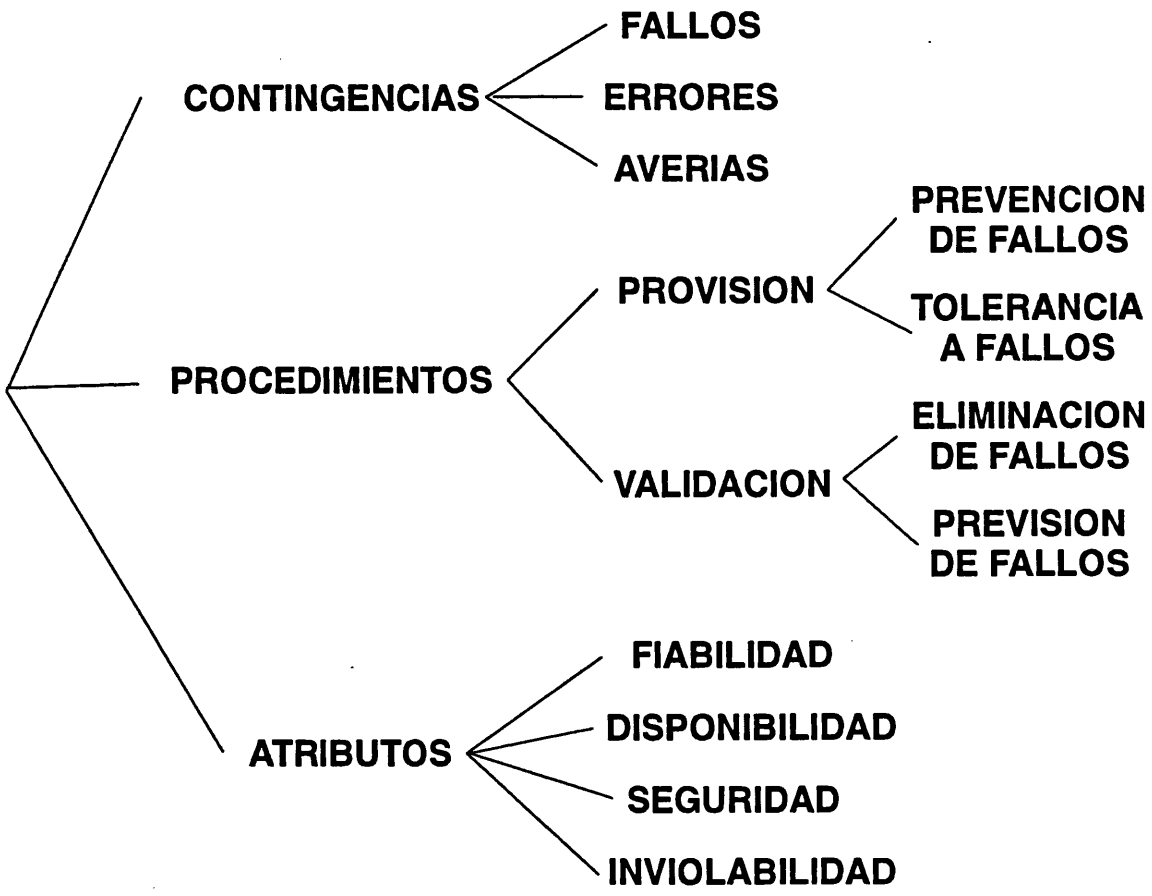


Fig. 1.1 *Taxonomía*

Los conceptos de garantía de funcionamiento en sistemas computacionales se pueden dividir en tres grandes categorías:

contingencias: se trata de circunstancias no deseadas que causan una desviación del funcionamiento correcto o son consecuencia de una desviación anterior; el resultado de una contingencia es que no se puede obtener el servicio esperado o que ya nunca más podrá ser obtenido.

procedimientos: en este grupo se engloban los métodos, herramientas y soluciones que posibilitan al sistema para el cumplimiento de sus servicios y permiten confiar justificadamente en la capacidad de hacerlo.

atributos o propiedades: conceptos que permiten expresar las propiedades esperadas del sistema y evaluar la calidad del sistema resultante de oponer los procedimientos a las contingencias.

1.1.1. CONTINGENCIAS

Toda definición de fiabilidad de un sistema debe incluir la distinción entre comportamientos adecuados e inadecuados del mismo. Para que esto sea posible, debe haber una clara especificación del comportamiento correcto del sistema [Nel87].

Avería (*failure*):

Una avería del sistema se produce cuando la respuesta a una solicitud de servicio se desvía de la respuesta esperada. A partir de esta consideración se concluye el siguiente principio: *"la fiabilidad de un sistema es inversamente proporcional a la frecuencia con que el sistema experimente averías"*.

Error:

cuando un módulo (sea hardware o software) del sistema produce una respuesta inadecuada asistimos a un error. Un error es la parte de un estado erróneo de un sistema que lo diferencia de un estado válido.

Fallo (*fault*):

se llama fallo a una condición presente en algún módulo del sistema, bien sea en el soporte físico (en adelante se empleará la palabra inglesa *hardware*) o en el soporte lógico (en lo sucesivo *software*).

Los fallos del hardware pueden deberse a factores físicos como perturbaciones externas, desgaste, errores de diseño, defectos de fabricación o limitaciones propias de los dispositivos. Los fallos del software sólo pueden ser debidos a errores humanos en el diseño o en la implementación.

Un fallo es, en sentido fenomenológico, la causa de un error. La aparición (o presencia) de un fallo lleva a un estado de latencia que puede conducir a un error o no hacerlo según las circunstancias que sobrevengan. La existencia de un error, si afecta a los servicios proporcionados por el sistema, puede llevar a una avería del mismo a menos que se adopten medidas de tolerancia a fallos. La avería de un sistema es causa de fallos en los sistemas que dependen de sus servicios. Esto constituye la cadena de contingencias

fallo → error → avería → [fallo → error → avería → [fallo → error → avería

La ocurrencia de un error puede deberse a tres causas: el módulo erróneo tenía presente un fallo; o el módulo ha recibido alguna entrada incorrecta; o bien se ha empleado inadecuadamente el módulo.

1.1.2. PROCEDIMIENTOS

Conseguir la garantía de funcionamiento de un sistema computacional implica la utilización conjunta de una serie de métodos que pueden clasificarse en [Lap92][Lee90]:

Prevención de fallos:

como evitar, por construcción, la aparición de fallos. Es aplicable a todo tipo de sistemas. El uso de componentes de alta calidad y el diseño extremadamente cuidadoso intentan prevenir la aparición de fallos.

Tolerancia a fallos:

como conseguir, mediante técnicas de redundancia, que los servicios del sistema se ajusten a las especificaciones aún en presencia de fallos. También es aplicable tanto a fallos en el hardware como fallos en el software.

Eliminación de fallos:

como minimizar, por validación, la presencia de fallos. Es especialmente aplicable a fallos en el software.

Previsión de fallos:

como estimar, por evaluación, la aparición y consecuencia de los fallos.

La prevención de fallos y la tolerancia a fallos son vistas como los procedimientos de garantizar el funcionamiento, esto es, procedimientos que conducen a poder depositar la confianza en los servicios que el sistema debe ofrecer. La eliminación de fallos y la previsión de los mismos son procedimientos de validación; procedimientos para justificar la confianza en los servicios.

Los procedimientos de eliminación de fallos están íntimamente ligados con los de prevención de fallos, y conjuntamente constituyen la **evitación de fallos** anterior a la fase de funcionamiento de un sistema (inicial o posteriores). No obstante, como se ha apuntado, este es un objetivo imposible de alcanzar plenamente; es más, sería aventurado pensar que no va a aparecer ningún fallo en la vida de un sistema.

Por tanto, la búsqueda de sistemas fiables en su actividad debe dedicar una adecuada cantidad de esfuerzos a la tolerancia a fallos, mediante el uso de determinadas técnicas de redundancia que se analizarán posteriormente.

1.1.3. ATRIBUTOS O PROPIEDADES

El cumplimiento de las garantías de funcionamiento de un sistema se intenta medir en términos probabilísticos como el grado y la manera en que el sistema se ajusta a sus especificaciones (al cumplimiento de los servicios esperados de él en las condiciones estipuladas).

La vida de un sistema se percibe por los usuarios según el estado del mismo en cuanto a cumplimiento de los servicios

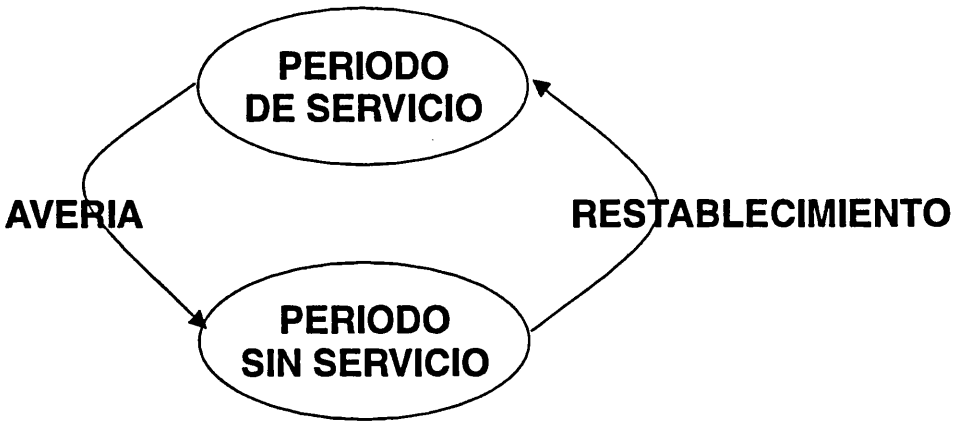


Fig. 1.2 *Periodos de vida de un sistema*

La cuantificación de estas transiciones llevan a tres atributos que expresan la capacidad del sistema de cumplir con sus servicios como funciones temporales [Nel87][Ying80]:

Fiabilidad (*reliability*) **R(t)**:

es la probabilidad de que un sistema no presente ninguna avería en el intervalo temporal de 0 a t, partiendo de un estado operativo en el instante t=0.

Disponibilidad (*availability*) **A(t)**:

la probabilidad de que un sistema funcione correctamente en el instante t.

Mantenimiento (*maintainability*) **M(t)**:

la probabilidad de que un sistema presente un estado operativo en el instante t habiendo estado averiado en el instante t=0.

Estas funciones temporales, que constituyen una estimación probabilística, suelen utilizarse para calcular otros parámetros de uso común más fácilmente interpretables, como son:

Tiempo medio hasta la avería MTTF:

valor esperado del intervalo de funcionamiento ininterrumpido.

Tiempo medio de la reparación MTTR:

valor medio del tiempo que tarda en volver al estado operacional después de una avería.

Tiempo medio entre averías MTBF:

se obtiene fácilmente como resultado de MTBF=MTTR+MTTF .

Disponibilidad esperada EA:

se define como la disponibilidad de un sistema reparable MTTF / (MTTF + MTTR).

Otros atributos, como son la seguridad o la inviolabilidad de un sistema, son atributos cualitativos de más difícil cuantificación. La propiedad cualitativa más importante trata de describir el efecto de las averías clasificando entre:

averías benignas, cuyas consecuencias son del mismo orden e magnitud que los beneficios obtenidos por el cumplimiento de los servicios en ausencia de fallos.

averías malignas, aquellas cuyas consecuencias son claramente más importantes que los beneficios obtenidos por el cumplimiento de los servicios en ausencia de fallos.

Esta clasificación involucra conceptos como el de **seguridad** *(safety)* de un sistema, como la propiedad de no producir consecuencias dañinas.

La medida de las propiedades descritas en este apartado suponen un estudio probabilístico no siempre concluyente que se basará en diversos modelos que se expondrán posteriormente. Los temas de seguridad de un sistema, como los de **inviolabilidad** *(security)* escapan al objeto de esta memoria de investigación por ser temas que requieren un tratamiento bien distinto.

1.2. TOLERANCIA A FALLOS

Considerando la imposibilidad de prevenir o eliminar todos los fallos, se centra la memoria en la tolerancia a fallos, por lo que se introducen algunas consideraciones al respecto [Nel87].

1.2.1. CARACTERIZACIÓN DE FALLOS

Los fallos pueden ser clasificados de muy distintas maneras, entre ellas:

tipo (o procedencia):

un fallo puede residir en el hardware o en el software;

causa:

determinar si diseño equivocado, fabricación incorrecta, causado externamente, desgaste de un componente, u otras causas;

modelo:

los fallos se suelen representar por modelos que intentan caracterizarlos;

duración:

un fallo es **permanente** cuando la causa no desaparece sin reparación explícita y el error es continuo; se dice que un fallo es **intermitente** cuando estando la causa presente el error sólo se produce a intervalos temporales circunstanciales; por último se llama fallo **transitorio** a aquel que desaparece (con sus efectos) sin haber sido reparado;

nivel:

un fallo del hardware puede estar localizado a nivel de un componente, de un módulo, de un subsistema o del sistema completo; los fallos del software pueden estar en un programa o en un microprograma;

extensión:

la extensión refiere al alcance de sus efectos pudiendo ir desde local al ámbito global;

latencia:

un fallo puede tardar un cierto tiempo en manifestar sus efectos (producción del error).

1.2.2. FASES DE LA TOLERANCIA A FALLOS

Todos los procedimientos de tolerancia a fallos, sean del tipo que sean, pueden estructurarse en cuatro fases, que, aplicadas consecutivamente, proporcionan la tolerancia. Estas cuatro fases son [Lee90]:

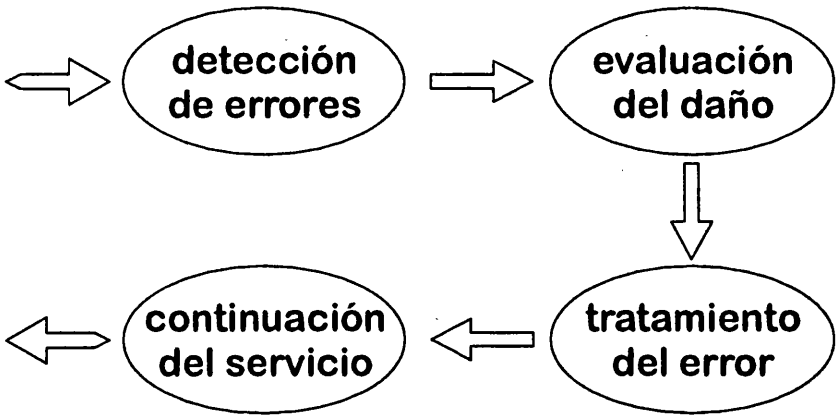


Fig. 1.3: *Fases de la tolerancia a fallos*

Detección de errores:

la mayoría de las estrategias necesitan algún medio de detectar la presencia de los fallos para adoptar medias protectoras y/o correctoras. Fenomenológicamente se produce la detección del fallo cuando aparece un error causado por aquél, esto es, el advenimiento de un estado erróneo que puede provocar una avería del sistema. La efectividad de las medidas detectoras de errores tiene una influencia directa sobre la eficiencia de la tolerancia a fallos.

Valoración del daño:

es necesario saber con la mayor exactitud posible las características del fallo causante del error (diagnóstico del fallo), y el alcance del mismo, esto es que partes del sistema se han desviado del funcionamiento correcto.

Tratamiento del error:

El estado erróneo del sistema debe ser transformado en un estado válido libre de error que permita continuar los servicios del mismo. Para ellos hay dos aproximaciones ampliamente empleadas

Recuperación por vuelta atrás: según esta aproximación, el proceso vuelve a un estado válido anterior (punto de recuperación, en ciertas aplicaciones llamado *checkpointing* por cuanto supone un punto de validez comprobada) para repetir los pasos seguidos (en las mismas o diferentes circunstancias). La recuperación por vuelta atrás puede ser empleada a muy distintos niveles y en prácticamente todo tipo de aplicaciones.

Recuperación hacia adelante: supone la continuación del servicio o proceso mediante la compensación de los errores, corrigiendo las desviaciones según un modelo establecido. Las acciones de compensación son fuertemente dependientes del tipo de aplicaciones, y no adaptables a cualquier proceso susceptible de fallos.

Continuación del servicio:

Antes de proceder a la continuación del servicio debe considerarse si debe tomarse alguna acción con respecto a la parte del sistema en que se ha detectado el error, como por ejemplo considerarla como fuera de servicio, o continuar simplemente el servicio sin adoptar ningún tipo de medida, esperando que no vuelva a manifestarse el error (caso de que se considere causado por un fallo transitorio o intermitente).

En caso de fallos transitorios o permanentes, puede decidirse no continuar el servicio. En tal caso puede hablarse de reparación, cuando se hace desaparecer el fallo del sistema, o de sustitución, cuando se reemplaza el sistema o parte fuera de servicio por otro en un estado válido. El restablecimiento, automático o por acción explícita, se considera como la vuelta a la provisión de servicios de un sistema que había estado fuera de servicio.

1.2.3. DECISIONES EN DISEÑO TOLERANTE A FALLOS

El primer paso del diseño de un sistema tolerante a fallos es el establecimiento de objetivos en cuanto a los fallos que van a ser tolerados y el grado deseado de fiabilidad, disponibilidad, o cualesquiera otros parámetros que se vayan a exigir. Por lo general se usa

una combinación de elusión y tolerancia de los fallos, pero los métodos empleados en cada caso dependen de las características de la aplicación y de los requerimientos que se soliciten [Avi75].

Por tanto, a la hora de iniciar el diseño debe tomarse una serie de decisiones de cara a la estrategia a seguir.

Redundancia protectora:

el uso de hardware suplementario, software añadido, duplicación de la información o repetición de las acciones para enmascarar los fallos o reconfigurar un sistema que ha presentado un fallo.

Enmascaramiento:

en ocasiones es suficiente con ocultar al exterior los efectos de un fallo durante un intervalo temporal sin necesidad de corregirlo.

Confinamiento de fallos:

es una tarea prioritaria el tratar de impedir que los efectos de un fallo se extiendan más allá del módulo en que se originan, tratando en cualquier caso que el alcance sea el menor posible. Por ello es comúnmente adoptada una arquitectura de bloques (descomposición modular) tanto del hardware como del software que permita confinar los fallos en el ámbito más cerrado que sea posible.

Diagnostico de fallos:

la identificación automática del módulo en que se produce el fallo y de la naturaleza del mismo es muy necesaria en sistemas de alta fiabilidad, así como en sistemas con reconfiguración. En muchos casos el diagnóstico se reduce al reconocimiento del módulo en que está encerrado el fallo.

Reparación y/o reconfiguración:

según el sistema, las partes erróneas pueden ser devueltas a la normalidad (*repair*) o descartadas definitivamente, debiendo darle una nueva estructura al conjunto (*reconfiguration*) poniendo en servicio los bloques suplementarios si los hubiere. En ambos casos, en el tiempo intermedio se puede producir una degradación de la eficiencia aunque intentando no interrumpir completamente la acción normal del sistema. También se intenta deshacer o corregir los daños producidos por el fallo.

Después de la reparación o reconfiguración el sistema debe ser devuelto a un estado de funcionamiento aceptable al máximo de sus posibilidades.

Degradación amistosa:

es la capacidad de un sistema para pasar a un estado operacional aunque degradado (funcionando bien pero con eficiencia menor de la usual) después de la ocurrencia de ciertos fallos.

El diseñador debe elegir cuales de estas capacidades son necesarias y suficientes para sus objetivos. Por ejemplo, en un sistema aeroespacial, un error instantáneo puede ser catastrófico y en consecuencia la aplicación debe enmascarar los errores. En general las técnicas a aplicar dependen tanto de la admisibilidad de errores como de la duración esperada de cada aplicación. Una aplicación crítica puede necesitar detección y recuperación automática para asegurar la disponibilidad del sistema durante la ejecución, pero en el caso de que la duración sea breve, el enmascaramiento puede ser suficiente. En los sistemas comerciales no críticos, una diagnosis de fallos extensiva unida a un adecuado confinamiento de los errores pueden ser suficientes para permitir una degradación amistosa y procedimientos manuales de reparación y recuperación.

1.3. TÉCNICAS DE REDUNDANCIA

La fiabilidad de un sistema se mejora por el uso de redundancia protectora a uno o más niveles dentro del mismo. Las técnicas de redundancia se pueden clasificar básicamente en tres grupos [Avi75]:

Redundancia espacial:

implica la existencia de dos o más réplicas de los módulos, componentes y procesos, que puedan ser elementos físicos, en cuyo caso se trata de redundancia del hardware, o elementos lógicos, en cuyo caso se trata de redundancia del software, o combinaciones de ambos.

Redundancia del hardware: se incluirá elementos adicionales para detección de los fallos, enmascaramiento, diagnostico o sustitución de módulos. En la detección se emplearán códigos de comprobación, temporizadores de guardia, módulos

replicados, comparadores [Hop78]. Para enmascaramiento se podrá usar redundancia triple con votadores .,etc...

Redundancia del software: se utilizará software extra para la detección, enmascaramiento, diagnóstico y tolerancia en general de fallos tanto del hardware como del software (aquellos para los que el sistema haya sido destinado). La redundancia en el software puede ser espacial (múltiples procesos ejecutándose simultáneamente) o temporal (ejecución secuencial de distintos procesos). [Ran75][Tay80][Hech79]

Redundancia de la información:

técnicas de codificación, detección y/o corrección usarán bits o bytes añadidos para intentar mantener la integridad de los datos, instrucciones o cualquier otra información necesaria o simplemente detectar cuando se produce alguna alteración indebida de los mismos.

Redundancia temporal:

distintas clases de operaciones (tanto del hardware como del software) y a distintos niveles podrán ser repetidas para sobreponerse a los efectos de fallos intermitentes o transitorios.

Los métodos de redundancia también se clasifican bajo otro criterio en redundancia estática o redundancia dinámica, según sea redundancia establecida de antemano o establecida durante la actividad del sistema. En cualquier caso, la redundancia puede utilizarse sólo para detectar errores, sólo para tolerarlos o para ambos.

Dominio / *Configuración*	**Redundancia Espacial**	**Redundancia de Información**	**Redundancia Temporal**
Estática	N-modular-redund. N-version-program.	Codificación Duplicación	Temporizadores Vuelta atrás
Dinámica	Repuestos Backups	Checkpointing	Recuperación

Fig. 1.4: *Tipos de redundancia*

Cada una de las técnicas empleadas tendrán unos beneficios en cuanto a tolerancia a fallos alcanzada, pero también llevarán aparejados un coste computacional y/o económico de modo que se deberá establecer un compromiso entre los diferentes factores en juego.

La descripción de las técnicas más empleadas contenida en los siguientes subapartados se hace desde el punto de vista hardware, aunque técnicas similares de redundancia se pueden aplicar al software. La generalización de estas técnicas tanto a hardware como a software se trata posteriormente.

1.3.1. MÉTODOS DE REPLICACIÓN

1.3.1.1. REDUNDANCIA N-MODULAR

La redundancia N-modular es en ocasiones conocida como redundancia masiva por la cantidad de hardware adicional empleado para su realización [Nel87][Ren78][Ins83]. Aunque costosa por este motivo, la redundancia N-modular (NMR por sus siglas inglesas) tiene la capacidad de enmascarar la presencia de un fallo sin necesidad de reconfiguración ni recuperación. Por ello, la tolerancia a fallos se consigue sin penalizar el coste temporal. Sin embargo, la cantidad extra de hardware reduce la fiabilidad del sistema a largo plazo. Además los fallos latentes enmascarados pueden acabar conduciendo a errores irreconocibles cuando se produzca un segundo fallo.

La materialización de una unidad NMR consiste en un conjunto de N módulos iguales que alimentan un elemento de salida V, que normalmente es un votador, aunque también puede realizar otras funciones como la selección de la mediana, por ejemplo. Cuando se usa como votador suele configurarse simplemente como una función de mayoría, aunque también puede plantearse como selector por mayoría cualificada. Si el umbral del votador es T, y N el número de módulos, la unidad NMR puede tolerar N-T averías de los módulos funcionales (aunque no la avería del votador, que es un punto crítico de la configuración).

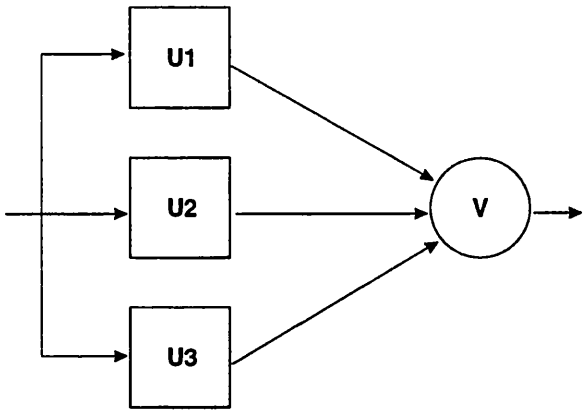


Fig. 1.5: *Redundancia Modular Triple (TMR)*

La redundancia triple modular (TMR) es la forma más sencilla de redundancia N-modular, donde los votadores son puertas tipo 2 de 3 y según lo dicho anteriormente toleran la avería de un módulo funcional.

El problema de la criticidad del votador puede resolverse con redundancia de votadores y de salidas, aunque, en cualquier caso, la interfaz final con el mundo físico, si la hay, debe ser única (en este sentido existen estudios sobre redundancia de sensores y redundancia de actuadores)

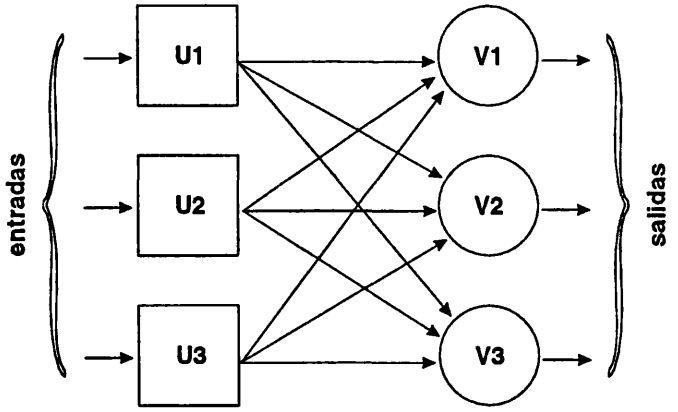


Fig. 1.6: *Redundancia Modular Triple (TMR) con votador redundante*

1.3.1.2. REPUESTOS

Los métodos que utilizan repuestos se conocen también como redundancia *stand-by* porque los módulos de repuestos suelen estar en un estado no operativo. El elemento de salida en estas técnicas es un simple selector que determina cual de las salidas de los módulos funcionales es la salida del conjunto.

El sistema debe disponer de los medios para detectar errores en los módulos funcionales. Los repuestos, que tienen utilidad ante una avería del módulo funcional primario, pueden estar activos o inactivos (durmientes) mientras no se les asigne el papel de módulo primario. Cuando los repuestos están activos, puede producirse la detección de errores por comparación. La duplicación es la configuración más sencilla para repuestos.

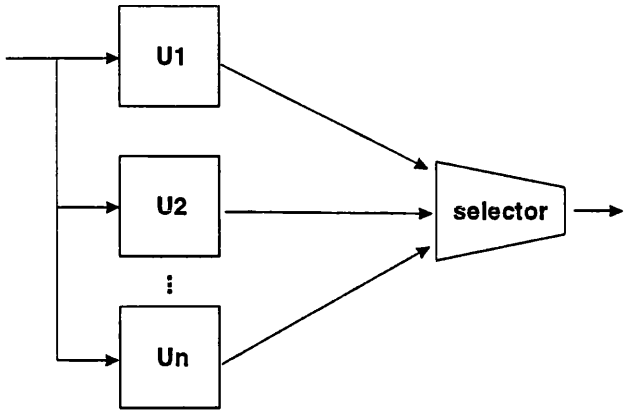


Fig. 1.7: *Redundancia con repuestos (sparing)*

1.3.1.3. SISTEMAS HÍBRIDOS

La redundancia híbrida consiste en combinar características de la redundancia N-modular con las de los sistemas con repuestos, de manera que utilizan como núcleo activo una unidad NMR manteniendo un número de módulos de repuesto que puedan sustituir los módulos averiados del núcleo NMR.

1.3.2. REDUNDANCIA DE LA INFORMACIÓN

La forma más difundida de redundancia es la utilizada en codificación donde bits o bytes suplementarios son añadidos a la información básica para permitir la detección y/o corrección de errores. La codificación también se utiliza para otros propósitos, como encriptado o compresión, pero este análisis de centra en el uso de la codificación para detección de errores y corrección de errores [Swe91].

La eficiencia de estos métodos se considera en términos de la cantidad de información total respecto a información básica y en términos de la sencillez de la implementación.

1.3.2.1. CODIFICACIÓN

Se llama mensaje a la información que debe codificarse mediante un código (la descripción se ciñe a códigos binarios). Sea k la longitud de un mensaje binario y M el conjunto de todos los posible mensajes binarios de longitud k (su cardinal es 2^k).

Sea n la longitud de una n-tupla binaria en la cual se pueden codificar los mensajes y sea U el conjunto de todas las posible n-tuplas. En el conjunto U, únicamente 2^k elementos de los 2^n que lo componen corresponderán a mensajes codificados. Sea C el conjunto de estos 2^k elementos. C es el llamado espacio de códigos y sus elementos son palabras del código. Los elementos de U-C no son palabras del código.

Sea X una palabra del código y sea Y una representación errónea de X motivada por un fallo. El error será detectable si Y no es una palabra del código. Esta situación se representa en la siguiente figura.

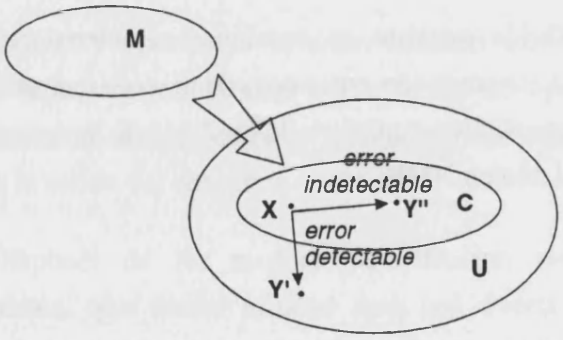


Fig. 1.8: *Errores detectables e indetectables en un espacio de códigos*

Se llama distancia de Hamming al número de componentes diferentes entre dos n-tuplas. Se conoce como mínima distancia de un código (espacio de código) C al mínimo de las distancias de Hamming entre todos los posibles pares de palabras del código.

Se dice que un código es separable cuando la información que se quiere codificar se puede extraer directamente de su correspondiente codificación sin necesidad de hacer ninguna transformación. Se dice que un código es sistemático cuando los procesos de codificación (aplicación de M a U) y descodificación (aplicación de U a M) pueden realizarse mediante operaciones aritméticas sobre las palabras manipuladas.

1.3.2.2. DETECCIÓN DE ERRORES

La capacidad de detección de errores de un código está relacionada con la distancia mínima del mismo. Un código C con una distancia mínima *dm* puede detectar hasta *dm-1* errores o alteraciones en una palabra. [Nel87][Chen84]

CÓDIGOS DE PARIDAD

Los códigos de paridad son los más antiguos, utilizados en aplicaciones de codificación de datos en memoria, en buses o en canales de comunicaciones. Tradicionalmente añaden un bit a un bloque de k bits de tal manera que en los k+1 bits que forman la n-tupla existan un número par (o impar) de valores diferentes de 0 (según se defina paridad par o impar), obteniéndose un código de distancia mínima 2 que permite detectar errores simples. La codificación y descodificación puede hacerse de manera sencilla con disposiciones de puertas XOR.

La cobertura de detección de errores por códigos de paridad puede ser mejorada mediante esquemas de paridad múltiple definidos sobre distintos subconjuntos de bits, llamados grupos de paridad. Los esquemas de paridad múltiple permiten también mejorar los diagnósticos respecto al origen del error.

CHECKSUMS

La técnica de *checksum* consiste en concatenar una cadena de s bits adicionales a un bloque de palabras de datos. El *checksum* más elemental de formar es la suma aritmética, módulo x, de las palabras de datos que forman el bloque. La técnica de *checksum* es sencilla de implementar para bloques de datos, tanto para codificación como para comprobación, pero permite una larga latencia de un fallo y es muy pobre para resolver diagnósticos cuando se detecta un error. Se utiliza frecuentemente en aplicaciones con dispositivos de almacenamiento secuencial, transferencias de bloques hacia o desde periféricos, memorias ROM, estructuras de datos y otras.

Un *checksum* de precisión simple se obtiene concatenando a un bloque de datos con longitud de palabra n un *checksum* de n bits calculado como la suma aritmética de todos los datos del bloque módulo 2^n. La capacidad detectora de errores está en función del tamaño del bloque de datos y del tamaño del *checksum*. Se pueden utilizar *checksum* extendidos utilizando más bits para el *checksum* n+a y realizando la suma modulo 2^{n+a}.

CÓDIGOS DE REDUNDANCIA CÍCLICA (CRC)

Los Códigos de Redundancia Cíclica o Códigos Polinómicos, consideran la secuencia de bits como la representación de un polinomio con coeficientes 0 y 1. La técnica consiste en añadir redundancia a la cadena de bits, mediante operaciones aritméticas, de forma que ésta sea divisible por un polinomio generador. Su uso más habitual es en la transmisión de datos (comunicación entre ordenadores). Si al dividir el polinomio que representa la trama recibida por el polinomio generador, se obtiene un resto no nulo, es debido a que ha ocurrido un error de transmisión. Los códigos CRC son especialmente útiles para la detección de ráfagas de errores, típicas de la transmisión de datos.

1.3.2.3. CORRECCIÓN DE ERRORES

Un código sistemático que codifica mensajes de longitud k en n-tuplas se conoce como código (n,k) y tiene una matriz generadora G de dimensiones $k \times n$, de manera que la palabra del código correspondiente al mensaje m se obtiene como $c = m.G$.

Un código con matriz generadora G tiene una matriz de comprobación H de dimensiones $(n-k) \times k$ tal que si c es una n-tupla del espacio de códigos entonces se cumple que $H.c^T = 0$.

En el caso de una n-tupla r correspondiente a la alteración de una n-tupla c del espacio de códigos produciéndose un error e, entonces $r = c + e$. Por ello, el resultado de aplicar la matriz de comprobación de paridad se conoce como síndrome y puede permitir identificar el error a partir del síndrome

$$s = H.r^T = H.(c+e)^T = H.c^T + H.e^T = 0 + H.e^T = H.e^T$$

De esta manera, la descodificación de un código r puede hacerse calculando el síndrome, determinando el error e producido a partir del síndrome de manera que a continuación se corrige el código erróneo r sumándole el error e para obtener el código correcto c (obsérvese que esta operación puede hacerse en cualquier caso, ya que si el código fuera correcto, el síndrome sería nulo y el error también, por lo que al sumarlo quedaría inalterado).

Un código de este tipo con distancia mínima *dm* es capaz de corregir t errores y detectar p errores adicionales si y solo si $2t + p + 1 <= dm$. La codificación, descodificación,

comprobación y corrección en estos códigos puede realizarse con circuitos relativamente simples.

1.3.3. REDUNDANCIA TEMPORAL

Las técnicas de tolerancia a fallos , sea a nivel hardware o a nivel software, que tratan de repetir una acción después de que haya fallado utilizan lo que se ha denominado redundancia temporal. Se describen a continuación las dos subclases de técnicas de tolerancia a fallos por redundancia temporal más desarrolladas: recuperación por vuelta atrás (*rollback-recovery*) y programación con bloques de recuperación.

1.3.3.1. VUELTA ATRÁS (*ROLLBACK-RECOVERY*)

En la técnica de recuperación por vuelta atrás [Chan75] los procesos guardan periódicamente información sobre su estado en almacenamiento estable mientras su funcionamiento está libre de fallos. Si se detecta algún error, los procesos utilizan la información guardada para restablecer un estado correcto anterior y reiniciar la ejecución desde dicho estado, en lugar de iniciar la computación desde el principio del proceso. Un estado correcto es aquel que se puede alcanzar durante la ejecución del proceso en ausencia de fallos.

Está técnica tiene la virtud de proporcionar tolerancia a fallos con bajos sobrecostes de supervisión y de recursos. Como recursos almacenamiento estable se suelen utilizar discos magnéticos. Los sobrecostes de supervisión en ausencia de fallos son los correspondientes al almacenamiento de la información de cada punto de recuperación. Además, si este almacenamiento y los mecanismos de vuelta atrás y reinicio se incluyen en el sistema operativo, la tolerancia a fallos resulta automática y transparente para las aplicaciones finales.

1.3.3.2. BLOQUES DE RECUPERACIÓN

La aplicación de la redundancia temporal para tolerar fallos del software [Hech79] se centra en la técnica de bloques de recuperación (el uso de replicación para fallos del

software se conoce como programación N-versiones, siendo el correspondiente en software de NMR). Cualquier programa progresa mediante acciones básicas elementales, que, obviamente, no pueden ser comprobadas una a una. Debe establecerse una adecuada frecuencia de comprobaciones para distribuirlas adecuadamente, agrupando las acciones básicas en bloques.

Un programa dividido en bloques considera cada uno de éstos como la unidad de detección de errores y recuperación, para lo cual cada bloque se acompañará con información extra que convierte a una agrupación de acciones básicas en un bloque de recuperación. Por tanto un bloque de recuperación lleva implícito un medio de detectar errores y cero o más alternativas.

1.4. MODELO IDEALIZADO DE COMPONENTE T.F.

Dentro de las metodologías más comúnmente empleadas en la provisión de tolerancia a fallos en sistemas de computación, prácticamente todos los intentos de una aproximación sistemática a la obtención de los objetivos de tolerancia pasan por establecer organizaciones claramente estructuradas de los sistemas, especificando las partes diferenciadas que los componen y la interacción entre las mismas, tanto desde el punto de vista del funcionamiento 'normal' de un sistema como desde los aspectos correspondientes a la tolerancia a fallos. [Lee90] Cada una de estas partes diferenciadas pueden ser a su vez descompuestas en partes más pequeñas, especificando igualmente sus interacciones, en un enfoque encaminado a la obtención de componentes más sencillos y más fácilmente tratables.

Este enfoque, denominado **descomposición modular**, simplifica el diseño, conduciendo a una estructura jerárquica en módulos y submódulos. Cada componente, a través de una interfaz de servicio, recibe solicitudes de otros componentes y las responde mediante su propia actividad interna que, en el caso de que el componente esté a su vez descompuesto en subcomponentes, hará uso de las facilidades provistas por algún subcomponente a través de la interfaz del mismo.

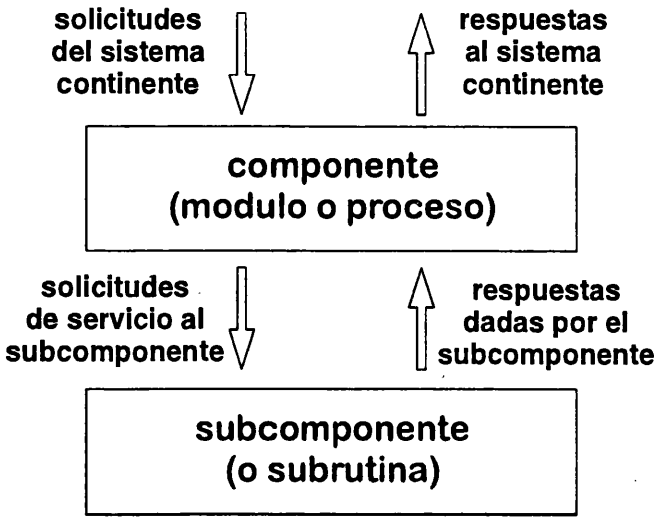


Fig. 1.9: *Enfoque por descomposición modular*

Esta descomposición modular, que es un método de diseño aplicable a cualquier sistema, incluso en aquellos que no incluyen la tolerancia a fallos entre sus objetivos, es especialmente aconsejable entre los sistemas que sí persiguen esta meta, y debe seguirse no sólo para las interacciones normales, llamando normal al funcionamiento de un sistema en ausencia de fallos, sino también en todas las acciones e interacciones que acontecen ante la aparición de un error (lo que se llamará funcionamiento anormal). Además, es deseable que, en un componente incluido en un sistema con provisión de tolerancia a fallos, se diferencien claramente las actividades correspondientes a un funcionamiento normal de las que corresponden al funcionamiento anormal.

Para mantener más claramente la separación entre actividades 'normales' y actividades 'anormales' algunos autores recomiendan la técnica del manejo de excepciones (*exception handling*) según la cual la detección de un error (a cualquier nivel) produce una excepción que automáticamente desvía la atención del funcionamiento normal invocando la rutina correspondiente al manejo de la excepción. La intención es que las restantes fases de la tolerancia a fallos, evaluación del error, recuperación del error y tratamiento del fallo, sean cubiertas por el gestor de la excepción, como se ha dicho separado del funcionamiento normal.

La combinación de ambas técnicas, descomposición modular y manejo de excepciones, sugieren el desarrollo de un sistema tolerante a fallos en base a componentes tolerantes a fallos ideales cuyo esquema de funcionamiento se expresa a continuación. [Lee90]

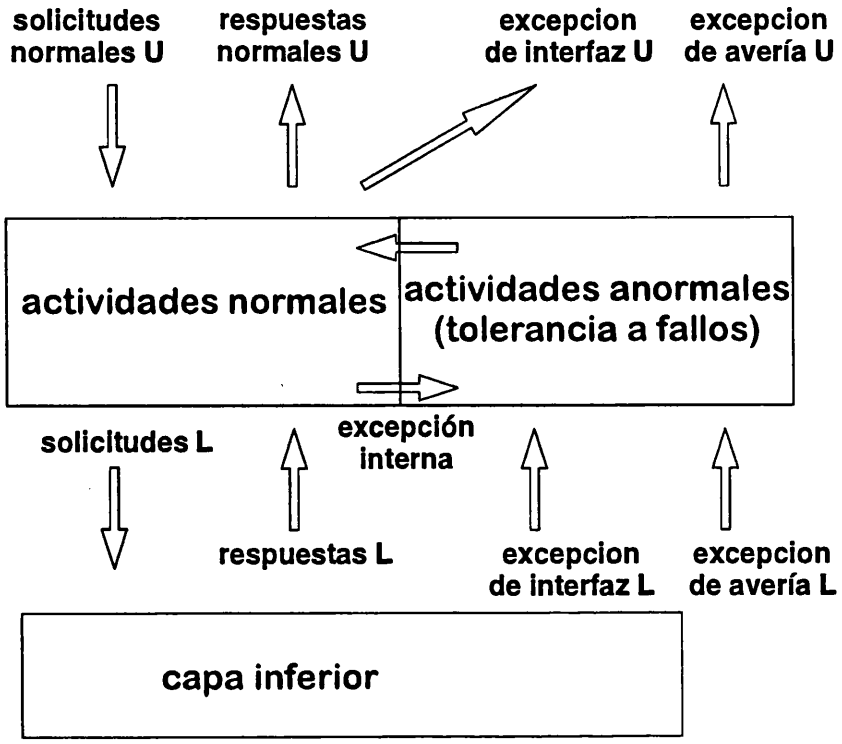


Fig. 1.10: *Componente idealizado tolerante a fallos*

La figura 1.10 representa un componente ideal que forma parte de un sistema que lo contiene. El componente en cuestión, dentro de su funcionamiento normal, recibe solicitudes de servicios provenientes del sistema continente (solicitudes de servicio U). Este, como parte de su actividad normal, provee el servicio solicitado y produciendo una respuesta (respuesta normal U). Si el componente está a su vez descompuesto en subcomponentes, puede requerir servicios a alguno de ellos (solicitudes de servicio L) y obtener respuesta de estos (respuestas normales L). Dentro del funcionamiento normal, el componente debe detectar solicitudes de servicio ilegales (supuestamente fruto de algún error) produciendo una excepción de interfaz (excepción de interfaz U).

El funcionamiento normal de un componente tolerante a fallos incluye también rutinas 'pasivas' (ejecutadas cuando se solicita un servicio al componente) o 'activas' (ejecutadas periódica o esporádicamente incluso en situaciones en que el componente no recibe solicitud alguna) para la detección de errores. Cuando una de estas rutinas para la detección de errores perciba la aparición de uno, generará una excepción interna que, como se ha indicado, invocará la actividad anormal por medio de un gestor de excepción (excepción interna).

La actividad anormal de un componente se iniciará, tal como se indica gráficamente, a través de tres tipos de excepciones: el primero tipo, la ya indicada excepción interna producida por la actividad normal del propio componente. Los otros dos tipos de excepciones provienen de subcomponentes (capas inferiores). El segundo tipo de excepción que conduce a la actividad normal es la interfaz de excepción proveniente de una capa inferior (excepción de interfaz L), fruto de una solicitud ilegal a un subcomponente. El último tipo de excepción se produce cuando un subcomponente es incapaz de tratar o recuperarse por si mismo de algún error, en cuyo caso la capa inferior notifica al componente una excepción de avería (excepción de avería L).

La actividad normal del componente tratará como se ha dicho, de evaluar el daño, recuperarse del error y tratar el fallo, teniendo en cuenta la información de la que disponga y el origen de la excepción (interna, de interfaz o de avería). En caso de conseguir aislar el error y mantener el componente en funcionamiento correcto, incluso en un funcionamiento aceptable aunque degradado, al finalizar la gestión de la excepción el componente vuelve a su actividad normal. Por el contrario, en caso de no poder contener el alcance del error o no poder garantizar el normal funcionamiento del componente, se notificará una excepción de avería del componente a la capa superior, esto es, al sistema que lo contiene (excepción de avería U).

Nótese que el sistema continente sólo será informado de los problemas del componente a través de una excepción de interfaz o una excepción de avería. En ningún caso, un problema de un subcomponente será trasladado directamente al sistema continente, sino que siempre será tratado, en primera instancia, por el componente. En caso de que el problema no sea resuelto, no se necesario que el sistema continente sepa de la naturaleza exacta del fallo, que puede residir en un subcomponente, siendo suficiente conocer que el componente no es capaz de proveer los servicios para los que fue establecido.

1.5. PROBLEMA GENERAL PLANTEADO

A continuación se establecerá una visión global procedimental de la tolerancia a fallos en los sistemas informáticos, estableciendo un enfoque general común tanto para fallos en el hardware como en el software y abstrayendo los mecanismos o recursos de tolerancia a fallos independientemente de que éstos sean físicos o lógicos.

El problema planteado es pues el hecho de que cualquier elemento en funcionamiento es susceptible de fallar produciendo efectos indeseables. Por ello, en la actualidad todo diseño ingenieril contiene tolerancia a fallos de sus elementos. En el proceso de especificación de estos sistemas es necesario definir cuándo el sistema está desarrollando correctamente su función, qué fallos debe ser capaz de tolerar y cómo debe hacerlo. Sin embargo, en la mayoría de las ocasiones, estos elementos de robustez y tolerancia a fallos no están incluidos en el diseño e implementación de una manera organizada que permita un análisis del cumplimiento de la fiabilidad exigible al sistema.

De hecho, como se ha expresado en apartados anteriores que introducían la fiabilidad de los sistemas, se aplicaban soluciones especificas para cada tipo de problema. Así cuando se trabajaba a nivel del hardware, se plantean una serie de soluciones [Hop78], aparentemente diferentes de las aplicadas a nivel del software [Kal93], aunque no conceptualmente diferentes [Ban94] como se explica posteriormente en este capítulo. Del mismo modo, en sistemas telemáticos se trata profusamente el control de errores [Swe91] que, con una terminología diferenciada de los sistemas tolerantes a fallos hardware o de los desarrollos de software tolerante a fallos, al fin y al cabo tratan de cumplir los mismos objetivos: cumplir fielmente con los servicios establecidos incluso ante la aparición de errores [ADKM92][BJ87].

1.5.1. ORGANIZACIÓN

En todos los casos se pueden expresar los procedimientos de tolerancia a fallos por medio de una organización en la que el elemento susceptible de fallar (sea hardware o software) se sitúa en una capa primaria que en principio realiza las funciones correspondientes al elemento, mientras que por encima de ésta tiene una capa de recubrimiento encargada de asegurar que el elemento cumpla con sus funciones correctamente o que al menos no produzca efectos indeseables.

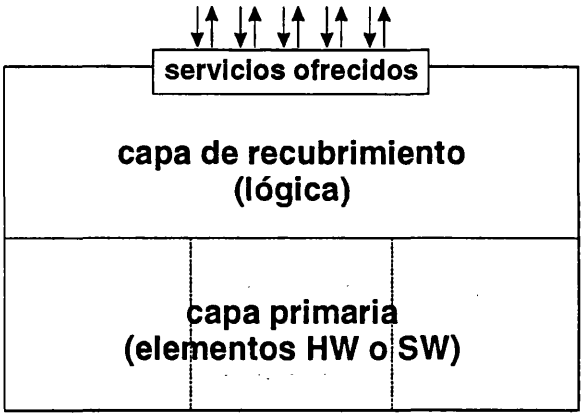


Fig. 1.11. *Organización general*

La capa primaria puede estar constituida por uno o más elementos, hardware o software, que atienden normalmente las solicitudes que reciben. La capa de recubrimiento será la que contenga los procedimientos de tolerancia a fallos, implementados en hardware o en software. En el resto de esta memoria se tratarán principalmente elementos con capas de recubrimiento puramente software.

1.5.2. CONSIDERACIONES SOBRE LA CAPA DE RECUBRIMIENTO

Sea cual sea el tipo de elementos de la capa primaria, la capa de recubrimiento debe estar condicionada por una serie de consideraciones

- La capa de recubrimiento debe incluir un sistema de recuperación integrado que conozca los recursos disponibles y los posibles fallos de los elementos de la capa primaria.

- Las actividades del sistema de recuperación integrado deben ser compatibles con un funcionamiento normal del sistema en ausencia de fallos.

- El sistema debe tener un sistema de monitorización y cuenta de alarmas para evitar que estas se den constantemente y establecer cuándo éstas superan los límites de lo permisible.

Cuando la capa de recubrimiento trabaje por procedimientos que precisan la repetición de alguna acción, deberá tener además las siguientes:

- La capa de recubrimiento es responsable de realizar una copia de un estado válido del funcionamiento del sistema (punto de recuperación) [Mul95].

- La capa de recubrimiento debe ser capaz de hacer que un elemento cualquiera de la capa primaria restaure sus condiciones de trabajo haciéndolas coincidir con las de un punto de recuperación guardado anteriormente [Chan75].

- La capa de recubrimiento debe asegurar que la repetición de una acción se realice siempre con las mismas condiciones. Para ello, en casos de simultaneidad y paralelismo de acciones, será necesario sincronizar los puntos de recuperación para evitar el llamado efecto dominó [BCS84][Nab93].

Por último, cuando la capa primaria se componga de más de un elemento realizando la misma función, o se produzcan repeticiones ordenadas por la capa de recubrimiento:

- La capa de recubrimiento debe asegurar que una acción repetida o duplicada no produzca duplicación de sus efectos

1.5.3. TERMINOLOGÍA EMPLEADA

Acciones básicas:

la consideración de acción básica se define para cada modelo, pudiendo ser desde una instrucción o una operación elemental, hasta un bloque de pasos, operaciones o instrucciones agrupadas según defina el modelo para cada sistema. El funcionamiento correcto del sistema supone la correcta ejecución de una serie de acciones básicas

Ejemplo de acciones básicas para cada modelo son una operación elemental (ciclo de ejecución) de un circuito digital, un bloque de instrucciones de un programa (bloque de recuperación) o la transmisión o recepción de un trama en comunicación de datos.

Junto con la definición de qué es una acción básica para cada modelo, habrá que distinguir entre

- acciones concatenadas: aquellas acciones que deben completarse correctamente antes de iniciar la siguiente y que si no son completadas correctamente suponen la suspensión de la serie de acciones.

- acciones no concatenadas pero imprescindibles: acciones que en caso de no ser completadas con éxito suponen la suspensión del conjunto de acciones básicas
- acciones no imprescindibles: aquellas que en caso de no ser completadas correctamente no suponen la suspensión del conjunto de acciones sino que se continúan a partir del siguiente punto de partida.

Puntos de recuperación y/o Punto de partida

estado definido por un conjunto de condiciones o variables que permiten, en caso de ser establecidos o restablecidos, la ejecución de una acción en las mismas condiciones que si se hubiera llegado a dicho punto mediante un funcionamiento correcto del sistema en la ejecución de las acciones.

Efecto de una acción:

modificación de las condiciones del sistema fruto de la ejecución o realización de la acción. Debe asegurarse la unicidad de efectos cuando trata de repetirse la ejecución de una acción.

Resultado de una acción:

cada acción debe producir un resultado comprobable (*self-checking*) que permita validar la corrección de la ejecución de dicha acción o un resultado comparable con los resultados de otras ejecuciones de la misma acción.

El resultado a considerar puede ser desde una variable o valor particular fruto del estado final de la acción hasta todo el conjunto de las condiciones finales de la realización de la acción.

Temporizadores:

para la ejecución de determinadas acciones, especialmente de acciones no elementales consistentes en un bloque de operaciones o instrucciones, se establece un tiempo máximo para la realización completa de la acción.

Recursos Disponibles:

se consideran recursos disponibles aquellos elementos, sean físicos o lógicos, que permiten la realización de un acción en condiciones normales.

La naturaleza de los recursos necesarios para la realización de acciones dependerá de la naturaleza de las mismas y del modelo empleado, pudiendo ser un circuito integrado, una unidad de proceso para la ejecución de un bloque de instrucciones de programa, un

algoritmo para la resolución de un problema o un canal de comunicación para acciones de transmisión de datos.

1.6. CLASIFICACIÓN DE LOS MODELOS

En este apartado se va a establecer una clasificación de los distintos modelos de tolerancia a fallos, tomando como primera base de la clasificación la cantidad de recursos disponibles, en segundo término el uso conjunto que se hace de dichos recursos y a continuación detalles particulares de cada disposición de recursos.

Disponibilidad de un sólo recurso

Recuperación hacia adelante (*Forward recovery*) (M1)

Recuperación por vuelta atrás (*Backward recovery*) (M2)

Disponibilidad de varios recursos

Utilización de varios recursos con la eficiencia de un único recurso

como Repuestos: un recurso primario y N-1 recursos alternativos

repuestos durmientes (M3)

repuestos activos (M4)

como Réplicas: múltiples recursos, comparando sus funcionamientos

réplicas (M5)

Utilización de varios recurso multiplicando la eficiencia de un único recurso:

sin repuestos ni replicas (M6)

con repuestos y/o réplicas (M7)

Para cada tipo de modelo se incluye

- descripción de la configuración en cuanto a recursos y uso de los mismos
- especificación de los requisitos necesarios para el modelo tratado
- exposición del procedimiento a seguir para los objetivos de tolerancia a fallos

- evaluación o comentarios sobre la idoneidad del modelo
- referencia a desarrollos existentes que correspondan al modelo.

1.6.1. DISPONIBILIDAD DE UN ÚNICO RECURSO

En los modelos en los que debe ejecutarse una acción disponiéndose de un único recurso para la realización de la misma, es absolutamente imprescindible la capacidad de VALIDACIÓN descrita que indique si se ha producido error.

En los dos modelos que se exponen en este apartado, recuperación hacia adelante y recuperación por vuelta atrás, el estado de AVERÍA se corresponde con la avería el único recurso disponible.

1.6.1.1. RECUPERACIÓN HACIA ADELANTE (M1)

DESCRIPCIÓN: MODELO 1

un único recurso disponible

REQUISITOS:

aplicable para acciones que necesariamente concluyen, cuyos resultados puedan ser validados y cuyos errores puedan ser corregidos.

PROCEDIMIENTO:

se realiza la EJECUCIÓN de cada acción mediante el único recurso disponible. La acción necesariamente llega a su fin y si la VALIDACIÓN indica que se ha producido error se intenta la CORRECCIÓN de los efectos erróneos. En caso de que la corrección no sea posible, se produce la AVERÍA. En todos los demás casos, el sistema prosigue con la ejecución de la siguiente acción.

COMENTARIOS:

la idoneidad de este modelo depende de la idoneidad y aplicabilidad de la VALIDACIÓN y de la CORRECCIÓN.

REFERENCIAS:

Este modelo corresponde al utilizado en las escrituras en memoria que utilizan códigos de Hamming, aunque en estos casos la CORRECCIÓN se hace sin necesidad de VALIDACIÓN, al aprovechar la propiedad de que la CORRECCIÓN de una codificación válida la deja inalterada, permitiendo extraer la información no redundante mientras que la corrección de una codificación errónea conduce a una codificación válida conteniendo la información adecuada [Swe91].

1.6.1.2. RECUPERACIÓN POR VUELTA ATRÁS (M2)

DESCRIPCIÓN: MODELO 2

un único recurso disponible

REQUISITOS:

aplicable para acciones que pudiendo concluir o no (en cuyo caso vence un temporizador), sus resultados puedan ser validados y sobre las que puede hacerse una RETROACCION para que pueden ser repetidos en caso de error, asegurándose la unicidad de efectos (el efecto de una nueva ejecución correcta después de cualquier número de repeticiones debe ser el mismo que si fuera la primera ejecución y esta hubiera sido correcta).

PROCEDIMIENTO:

se inicia la EJECUCIÓN de la acción mediante el único recurso disponible. Si la acción llega a su fin antes de que venza el temporizador se procede a la VALIDACIÓN de sus resultados. En caso de que se detecte un error, se procede a la RETROACCIÓN y se inicia nuevamente la ejecución de la acción mediante el único recurso disponible.

Si ha vencido el TEMPORIZADOR antes de que se complete la acción, se detiene la acción si es posible y también se procede a la RETROACCIÓN y la repetición de la ejecución. Debe asegurarse la unicidad de efectos de las repeticiones tanto para las repeticiones por validación con resultado negativo como por las repeticiones por vencimiento de temporizadores.

Se establece un número máximo de repeticiones, contando conjuntamente las repeticiones por uno u otro motivo, de tal manera que si se consume el número máximo

establecido sin haber logrado una ejecución correcta de la acción se produce la AVERÍA del sistema.

COMENTARIOS:

la idoneidad de este modelo depende de la idoneidad de la VALIDACIÓN. La latencia de un fallo permanente o de larga duración dependerá del número máximo de intentos permitidos que se establezca.

REFERENCIAS:

Este modelo corresponde al utilizado en ARQ (*Automatic Request for Replay*) en transmisión de datos, que se describirá con mayor detalle en el apartado correspondiente del capítulo dedicado al problema de las comunicaciones, al centrarse esta memoria de investigación en la aplicación sobre ese campo [Bla93][Swe91].

1.6.1.3. COMPARACIÓN Y LIMITACIONES

Ambos modelos necesitan un medio eficaz de VALIDACIÓN de los resultados por sí mismos (*self-checking*)

El primero (M1) requiere medidas correctoras especiales, que además tienen una posibilidad de fallo. Debe considerarse también el tiempo empleado en las medidas correctoras. Como contrapartida positiva, el advenimiento de la avería ante la presencia de un fallo irreparable es casi inmediata.

El segundo modelo (M2) requiere sobre todo tiempo (por lo cual la eficiencia es más baja en caso de errores frecuentes), pero no requiere de complicadas medidas de corrección. Su aspecto más complicado es la necesidad de asegurar la unicidad de efectos de las acciones repetidas. La latencia de un fallo permanente o de larga duración depende del número N de intentos permitidos.

Como limitación principal obvia de ambos modelos debe expresarse que ninguno de los dos modelos tolera fallos permanentes del recurso.

1.6.2. MULTIPLICIDAD DE RECURSOS CON EFICIENCIA DE UNO

Dentro de los modelos en que se emplean varios recursos, puede utilizarse esta multiplicidad para cumplir la función de un único recurso libre de fallos, de manera que la multiplicidad es empleada únicamente para conseguir tolerancia a fallos.

En cualquier caso estos recursos disponibles pueden ser iguales o diferentes.

Esta multiplicidad de recursos puede organizarse de diferentes maneras según se consideren todos los recursos a un mismo nivel (réplicas) o se considere un recurso como recurso principal y el resto como recursos secundarios o alternativos (repuestos).

1.6.2.1. REPUESTOS

Se considera que uno de los recursos es el recurso principal que se emplea para la ejecución de la acción en ausencia de fallos, mientras que los otros recurso sólo tienen papel en caso de que aparezca un fallo en el funcionamiento del recurso principal. Los resultados de las acciones se comprueban por VALIDACIÓN (*self-checking*) o por COMPARACIÓN en los casos que es posible.

1.6.2.1.1. REPUESTOS DURMIENTES (M3)

DESCRIPCIÓN: MODELO 3

uno de los recursos es el recurso principal que permite la realización de las acciones. Los demás recursos (N-1) no realizan ninguna acción mientras no haya fallos en el principal y son los recursos de repuesto.

REQUISITOS:

aplicable para acciones que pudiendo concluir o no (en cuyo caso vence un temporizador), sus resultados puedan ser validados y sobre las que puede hacerse una RETROACCION para que pueden ser nuevamente ejecutados mediante el uso de otro recurso, asegurándose la unicidad de efectos.

PROCEDIMIENTO:

se inicia la EJECUCIÓN de la acción mediante el recurso principal. Si la acción llega a su fin antes de que venza el temporizador se procede a la VALIDACIÓN de sus resultados.

En caso de que se detecte un error, se procede a la RETROACCIÓN al punto de recuperación, se designa uno de los recursos de repuesto como recurso principal, se realizan las acciones necesarias para el CAMBIO DE RECURSO, para a continuación iniciar nuevamente la ejecución de la acción mediante el recurso principal en vigor.

Si ha vencido el TEMPORIZADOR antes de que se complete la acción, se detiene la acción si es posible y también se procede a la RETROACCIÓN y al CAMBIO DE RECURSO y la realización alternativa de la ejecución. Debe asegurarse la unicidad de efectos de las ejecuciones alternativas.

El número de alternativas viene dado por el número de recursos de repuesto disponibles, agotados los cuales se produce la AVERÍA del sistema. En caso de producirse un error en la ejecución de la acción mediante un recurso, debe diagnosticarse la duración del fallo que los produce y decidir si el recurso es inhabilitado y no tenido en cuenta para la ejecución de posteriores acciones o permanece en uso como recurso de repuesto.

COMENTARIOS:

El punto específico a tratar en una configuración con repuestos durmientes es el CAMBIO DE RECURSO, que permita que la ejecución de la acción desde el punto de partida mediante el nuevo recurso tenga los mismos resultados y efectos que hubiera tenido en el recurso anterior. La eficiencia en tiempo de ejecución disminuye fuertemente con el crecimiento de la tasa de errores, igual que en el modelo 2.

REFERENCIAS:

Este modelo es el empleado en tolerancia a fallos en el software por bloques de recuperación que establecen para un grupo de instrucciones de programa un punto de recuperación al inicio, un módulo alternativo primario, N-1 módulos alternativos secundarios, un test de aceptación y un procedimiento de volver al punto de recuperación al inicio si el test de aceptación es negativo [Hech79][Chan75].

1.6.2.1.2. REPUESTOS ACTIVOS (M4)

DESCRIPCIÓN: MODELO 4

uno de los recursos es el recurso principal que permite la realización de las acciones. Los demás recursos (N-1) realizan la misma acción que el recurso principal debiendo coordinarse las diferentes ejecuciones.

REQUISITOS:

aplicable para acciones que pudiendo concluir o no (en cuyo caso vence un temporizador), sus resultados pueden ser validados o comparados y que pueden coordinarse o sincronizarse diversas ejecuciones de la misma acción mediante varios recursos, de manera que los resultados y efectos de la acción sean únicos.

PROCEDIMIENTO:

se inicia la EJECUCIÓN de la acción simultáneamente mediante todos los recursos disponibles en espacios de acción separados a partir de unas mismas condiciones de punto de partida. Si ha vencido el TEMPORIZADOR antes del final de la ejecución de la acción mediante el recurso primario o ésta ejecución llega a su fin y la VALIDACIÓN de los resultados indica que se ha producido error, entonces se comprueba si algunas de las ejecuciones alternativas se ha completado correctamente. En tal caso los resultados y efectos de dicha ejecución se tomarán (estableciendo los mecanismos necesarios) como resultados y efectos de la acción. Esto puede hacerse por orden preestablecido, si se ha asignado un orden determinado a los repuestos, o por orden temporal, tomando la primera ejecución correcta que se produzca en el tiempo. Si ninguna de las ejecuciones de la acción es completada correctamente antes de que venza su temporizador, entonces se produce la AVERÍA del sistema.

Una alternativa para el caso de repuestos activos es comparar los resultados de la ejecución mediante el recurso principal con los de una de las ejecuciones de repuesto. En caso de no coincidir, se abre un abanico de posibilidades para determinar la selección, según el número de repuestos disponibles.

En caso de producirse un error en la ejecución de la acción mediante un recurso, debe diagnosticarse la duración del fallo que los produce y decidir si el recurso es inhabilitado o permanece en uso como recurso de repuesto.

COMENTARIOS:

El problema principal de esta configuración de repuestos activos es la necesidad de coordinación de las distintas ejecuciones para que cuenten con un mismo punto de partida pero produzcan efectos sobre espacios de acción diferentes, y que se tome como efecto real de la acción sólo el de una de las ejecuciones (selección).

La avería final sólo se produce cuando fallan todos los recursos, y si debe producirse se produce en el tiempo necesario para determinar la avería de uno de los recursos.

REFERENCIAS:

Una configuración típica es la duplicación, donde se comparan los resultados del recurso primario con el recurso de repuesto para detectar errores. Para tolerar fallos es más utilizada la configuración de par con repuesto (*pair-and-spare*) donde se utilizan dos recursos activos y uno de repuesto durmiente, de tal manera que cuando se detecta una diferencia entre los resultados de las ejecuciones de una acción en los dos recursos activos, se pasa a realizar ésta mediante el tercer recurso (el recambio) [Nel87].

1.6.2.1.3. COMPARATIVA REPUESTOS DURMIENTES Y ACTIVOS

La diferencia entre repuestos durmientes y activos es que mientras con los repuestos durmientes la gestión es más sencilla (sólo se produce una ejecución de la acción en cada momento) el tiempo de realización es en consecuencia mayor en presencia de fallos. Por contra, con los repuestos activos el tiempo de ejecución es menor con más baja incidencia de la presencia de fallos, pero el problema de coordinación es en ocasiones excesivamente complejo.

1.6.2.2. RÉPLICAS (M5)

Cabe diferenciar aquí las réplicas de los repuestos en el hecho de que al hablar de réplicas se indica que todos los recursos están activos y que no existe un recurso primario junto a un conjunto de recursos alternativos, sino simplemente un conjunto de recursos de igual importancia.

DESCRIPCIÓN: MODELO 5

se dispone de un número de recursos iguales no inferior a tres que permitan la realización simultánea de las acciones, produciéndose una selección de los resultados de todos ellos.

REQUISITOS:

aplicable para acciones cuyos resultados (en caso de completarse las acciones) pueden ser comparados y que pueden coordinarse o sincronizarse diversas ejecuciones de la misma acción mediante varios recursos, con capacidad de seleccionar mediante un votador los resultados y efectos de una de ellas.

PROCEDIMIENTO:

se inicia la EJECUCIÓN de la acción simultáneamente mediante todos los recursos disponibles en espacios de acción separados a partir de unas mismas condiciones de punto de partida. Conforme se van completando las ejecuciones se realiza la COMPARACIÓN de las resultados. Las ejecuciones en las que vence el TEMPORIZADOR antes de completarse no son tenidas en cuenta.

Un dispositivo o algoritmo votador produce como salida los efectos de una ejecución tales que los resultados se repitan en un número mínimo (umbral) de réplicas. Así por ejemplo, en sistemas con 3 réplicas, el votador se establece con un umbral igual a 2.

Si el máximo número de resultados de ejecuciones que se completen en el término establecido antes del temporizador y sean iguales entre sí es menor que el umbral del votador entonces se produce la AVERÍA del sistema.

En caso de que el votador seleccione un grupo de resultados iguales, los recursos cuyos resultados suponen que han sufrido un fallo, aunque se mantienen activos. Sin embargo, un recursos que repetidamente produce resultados discordantes puede suponerse que sufre un fallo permanente y proceder en consecuencia.

COMENTARIOS:

En las configuraciones de réplicas con votadores, la tolerancia a fallos es inmediata, permitiendo tolerar un número de errores máximo de N-U siendo N el número de réplicas y U el umbral del votador (N suele ser un número impar y U suele ser la parte entera de (N+1)/2). Como contrapartida, la latencia de los fallos puede prolongarse, al funcionar por enmascaramiento, no detectándose hasta que se produce un fallo masivo.

REFERENCIAS:

Este modelo corresponde al conocido de redundancia N-modular (programación de N versiones en el caso del software) [Hop78][Avi85].

1.6.2.3. COMPARATIVA REPUESTOS Y RÉPLICAS

Como ya se ha apuntado, el enmascaramiento propio del modelo 5 permite una tolerancia casi inmediata, con el peligro de latencia de fallos no detectados. Los modelos con réplicas requieren en cualquier caso un número mayor de recursos.

1.6.3. MULTIPLICIDAD PARA EFICIENCIA

La disponibilidad de múltiples recursos puede también aprovecharse para mejorar la eficiencia del sistema cuando diversas acciones puedan ser ejecutadas simultánea o concurrentemente mediante los diferentes recursos. Esto que es la base de la computación paralela, es también aprovechable en otras áreas de aplicaciones, como la comunicación de datos, siempre que sea posible esta paralelización de acciones.

De esta manera, los recursos pueden ser utilizados de diferentes maneras: todos los recursos para la realización de acciones, un conjunto de recursos para la realización de acciones y otros recursos como repuestos, diversas agrupaciones de réplicas para la ejecución de tantas acciones como agrupaciones de réplicas, u otras configuraciones.

1.6.3.1. SIN REPUESTOS NI REPLICAS (M6)

DESCRIPCIÓN: MODELO 6

se dispone de un número de recursos que permiten la ejecución simultánea de acciones en espacios de acción diferentes. Todos los recursos se emplean activamente para la realización de acciones diferentes.

REQUISITOS:

aplicable para procesos con acciones paralelizables que puedan ejecutarse simultánea o concurrentemente con ASIGNACIÓN a distintos recursos y donde la ejecución de cada acción pueda ser VALIDADA por sí misma a partir de sus resultados y sobre la que puedan efectuarse tanto RETROACCIÓN como CAMBIO DE RECURSO.

PROCEDIMIENTO:

se van iniciando las EJECUCIONES de las acciones en los distintos recursos disponibles, por un criterio de asignación que se establezca. Cuando la ejecución de una acción en el recurso al que haya sido asignada concluye incorrectamente o vence su temporizador, se produce la RETROACCIÓN para intentar repetidamente la ejecución.

Si se agota el número máximo de repeticiones establecido, se produce la AVERÍA del recurso y se procede al CAMBIO DE RECURSO según el criterio de asignación para iniciar la ejecución de la acción nuevamente mediante otro recurso. Por cada avería de un recurso, se produce una degradación de la eficiencia del sistema, que dispone de un recurso menos para la ejecución de las acciones.

El criterio de asignación debe tener en cuenta los recursos que están en estado de AVERÍA para no asignarles acciones a ejecutar. La AVERÍA total del sistema se produce cuando todos los recursos del mismo están en estado de AVERÍA.

COMENTARIOS:

Esta configuración multiplica la eficiencia en ejecuciones paralelas en ausencia de fallos con lo cual utiliza óptimamente los recursos si la carga del sistema lo exige.

1.6.3.2. CON REPUESTOS Y/O RÉPLICAS (M7)

DESCRIPCIÓN: MODELO 7

se dispone de un número de recursos que permiten la ejecución simultánea de acciones en espacios de acción diferentes. Sin embargo no todos los recursos se emplean activamente para la realización de acciones diferentes, sino que se dispone de un conjunto de recursos para usar como repuestos o réplicas de los activos (según se ha descrito en los modelos 3, 4 y 5)

REQUISITOS:

aplicable para procesos con acciones paralelizables que puedan ejecutarse simultánea o concurrentemente con ASIGNACIÓN a distintos recursos y donde la ejecución de cada acción pueda ser VALIDADA por sí misma o COMPARADA con otras a partir de sus resultados y sobre la que puedan efectuarse tanto RETROACCIÓN como CAMBIO DE RECURSO.

PROCEDIMIENTO:

se van iniciando las EJECUCIONES de las acciones en los distintos recursos activos, por un criterio de asignación que se establezca. Si se utilizan repuestos activos o réplicas algunas acciones se ejecutarán simultáneamente en dos o más recursos. Cuando se detecte un error (por VALIDACIÓN, COMPARACIÓN o vencimiento del TEMPORIZADOR) puede intentarse un número de reintentos sobre el mismo recurso o agrupación de ellos o determinar la AVERÍA del recurso y proceder al CAMBIO DE RECURSO para iniciar nuevamente la acción.

En este modelo el número de recursos efectivos para la realización de acciones es N y tienen capacidad de validación autónoma, y el número total de recursos disponibles es T,

las T-N primeros averías de recursos no suponen ninguna degradación del sistema, empezando a perderse eficiencia a partir de la avería T-N+1.

El criterio de asignación tendrá en cuenta los recursos que están en estado de AVERÍA para no asignarles acciones a ejecutar. La AVERÍA total del sistema se produce cuando todos los recursos del mismo están en estado de AVERÍA.

COMENTARIOS:

Esta configuración multiplica la eficiencia en ejecuciones paralelas en ausencia de fallos reservando recursos para la tolerancia a fallos. Para sistemas con mucha carga de trabajo, la tolerancia supone una merma de la eficiencia de tal manera que debe establecerse un compromiso entre ambas [Nel87][Bri94].

1.6.3.3. COMPARATIVA

Entre los dos grandes modelos presentados para multiplicidad de recursos y concurrencia de acciones diferentes (modelos 6 y 7) la diferencia substancial estriba en que el modelo 6 se basa en la suposición optimista de que los fallos van a ser pocos y directamente se aprovechan todos los recursos para incrementar la eficiencia, que disminuye en presencia de fallos, mientras que el modelo 7 prevé anticipadamente los fallos y no confía en alcanzar la máxima disponibilidad de los recursos. Por otro lado, algunos sistemas no tienen suficiente carga o no la pueden paralelizar como para ocupar todos los recursos, luego en tales casos el mantener unos recursos inactivos no supone ningún perjuicio a la eficiencia global.

1.7. RELACIONES Y TRANSICIONES ENTRE MODELOS

Las configuraciones que utilizan múltiples recursos pueden cambiar, fruto de averías de algunos de dichos recursos, de un modelo de funcionamiento a otro. Así por ejemplo, un sistema de tres réplicas con votador (modelo 5) puede pasar a funcionar como un sistema de par y repuesto (modelo 7) según las condiciones de funcionamiento, o si se considera que alguno de los recursos presenta una frecuencia de fallos elevada pasar a utilizarlo como repuesto durmiente (modelo 3) y funcionar con una configuración de recurso primario (uno de los dos que funcionan correctamente) y recursos secundarios.

De igual modo, una configuración según el modelo 7 en la que se produce la avería de un recurso, puede seguir dentro de los esquemas del modelo 7 hasta T-N averías de recursos, en cuyo caso es idéntico al modelo 6 (todos los recursos se utilizan eficientemente y no hay ninguno como repuesto). Por otro lado, tanto en el modelo 6 como en el modelo 7, ante la aparición de fallos puede decidirse dedicar más recursos a la tolerancia que a la eficiencia pudiendo funcionar como un único recurso tolerante como en los modelos 3, 4 y 5 si se quiere asegurar el funcionamiento continuo y fiable [Bri94][Bri95] aun a costa de la eficiencia.

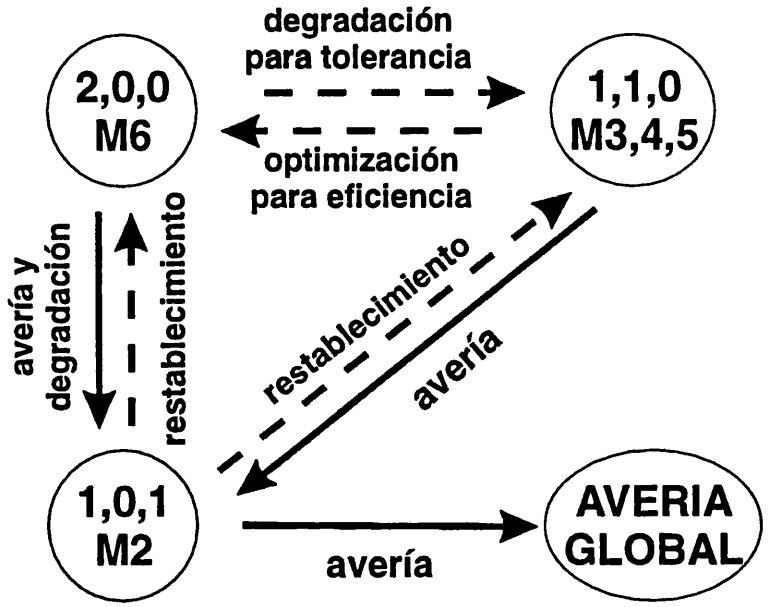


Fig. 1.12: *Relaciones y transiciones con dos recursos disponibles*

En las figuras 1.12 y 1.13 se muestran las posibles opciones y transiciones en un sistema con dos y tres recursos respectivamente, según una notación (N,S,D) donde se indica con N el número de recursos empleados eficientemente, con S el número de recursos de repuesto (o réplicas) y D el número de recursos averiados (nivel de degradación).

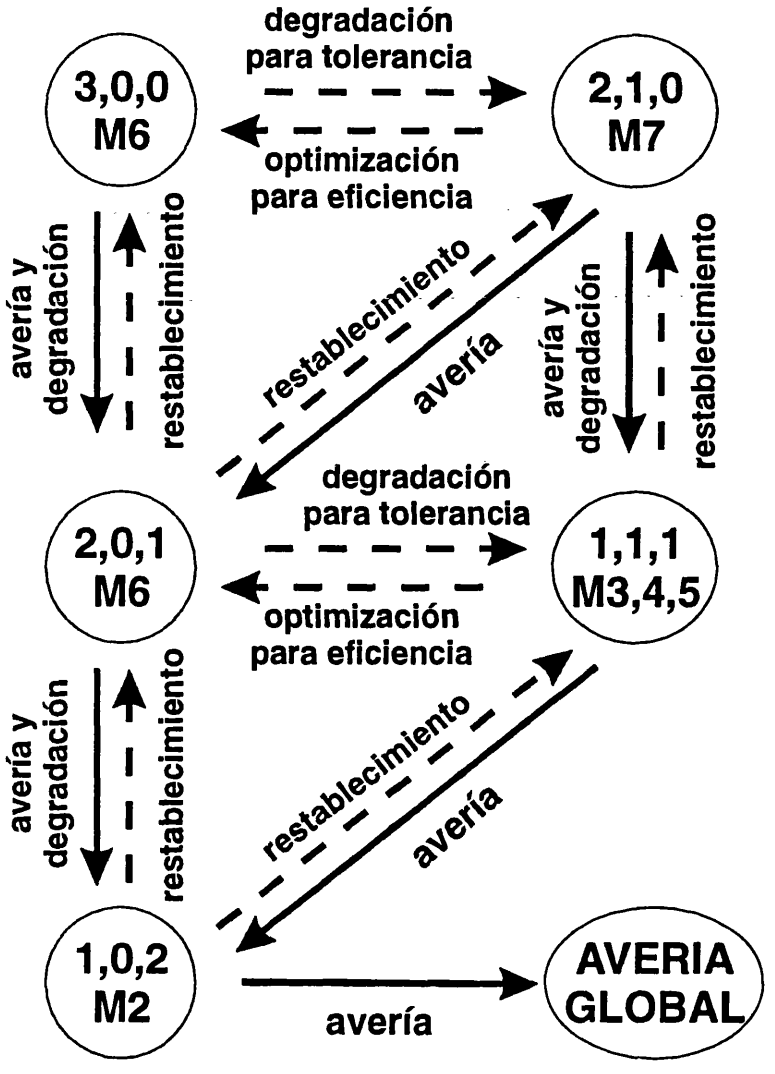


Fig. 1.13: *Relaciones y transiciones con tres recursos disponibles*

1.8. CONCLUSIONES

Las especificaciones del diseño de un sistema de control deben incluir no sólo las condiciones del funcionamiento sino también las circunstancias que pueden afectar al mismo causando desviaciones en su comportamiento y el modo de detectar dichas desviaciones.

Igualmente, el diseño del sistema debe establecer los atributos que debe cumplir el mismo en cuanto a fiabilidad, disponibilidad y seguridad, determinando los métodos y herramientas que posibiliten al sistema la consecución de dichos atributos teniendo en

cuenta las contingencias que puedan incidir en el funcionamiento del mismo. En general las técnicas a aplicar dependen tanto de la admisibilidad de errores como de la duración esperada de cada aplicación. Una aplicación crítica puede necesitar detección y recuperación automática para asegurar la disponibilidad del sistema durante la ejecución, pero en el caso de que la duración sea breve, el enmascaramiento puede ser suficiente. En sistemas comerciales menos críticos pero de más larga duración, una diagnosis de fallos extensiva unida a un adecuado confinamiento de los errores pueden ser suficientes para permitir una degradación amistosa y procedimientos manuales de reparación y recuperación.

En caso de que los requisitos de fiabilidad de un sistema justifiquen la existencia de recursos redundantes, el diseño debe definir cuál va a ser el uso de dichos recursos redundantes. Para ello, se han presentado en este capítulo una serie de modelos de gestión de los recursos redundantes, comparando unos modelos con otros según el tipo de sistema que se trate y sus necesidades. Además, y como se ha expresado en el último apartado, la selección de uno de estos modelos puede hacerse de modo dinámico, según las condiciones del sistema. En consecuencia, el diseño de un sistema de control con recursos redundantes determina el modelo de tolerancia a fallos que le permita cumplir sus requisitos de fiabilidad y los cambios en la configuración del sistema (transiciones entre modelos) que puedan producirse según las contingencias acaecidas.

CAPÍTULO SEGUNDO

COMUNICACIONES EN ITS

Según datos de los últimos estudios encargados por la Unión Europea [EC96], el transporte de mercancías por carretera supone el 70% del total del transporte de mercancías en el área de la Unión Europea, mientras el segundo medio de transporte de mercancías es el ferrocarril, cuyo volumen de transporte es inferior al 16% del total. Igualmente, se estima que el 88% de los desplazamientos interurbanos de personas en el marco de la Unión Europea se realizan por carretera, utilizándose transporte privado en la mayor parte de los mismos.

La creciente utilización del transporte por carretera en la Unión Europea y el consiguiente problema de la congestión del tráfico prôduce:

a) un incremento de los retrasos en los desplazamientos de personas y mercancías,

b) un riesgo para la seguridad personal,

c) un coste mayor de energía,

d) un deterioro medioambiental y

e) una escalada en la frustración diaria de los afectados.

Todo ello supone en conjunto un impacto socioeconómico de gran importancia en la actualidad. De hecho, los estudios de la Unión Europea estiman los costes de la congestión del tráfico en 125.000 millones de dólares por año y el coste en vidas humanas del transporte por carretera en 44.000 muertes por año en la U.E. debiendo incrementar a estos números el efecto del deterioro medioambiental de difícil cuantificación.

Un informe del Texas Transportation Institute sobre la movilidad en cincuenta áreas urbanas de los Estados Unidos de América (Los Ángeles, Washington D.C., Miami) [TTI94] estimó el coste total de la congestión del tráfico en las áreas estudiadas durante

1992 (retrasos más combustible malgastado) en 48 mil millones de dólares (un 89% del total atribuible a los retrasos), con un incremento del 9% respecto a la estimación de 1991, y con una "tasa de congestión" máxima de 820 dólares por persona y 1580 dólares por vehículo en Washington D.C. Además, pese al incremento de los retrasos debidos a las congestiones, el 76% de los trabajadores que se desplazan por carretera para ir a trabajar lo hacen solos en su vehículo, mientras un 12% utiliza 'auto compartido' y sólo un 5% utiliza el transporte colectivo. Aun más preocupante es la estimación de 40.000 muertos en accidentes de tráfico durante 1993, sobre todo teniendo en cuenta que es un efecto generalmente asumido cuando el uso de la tecnología podría salvar la mayoría de ellas.

Además de otros recursos relacionados con un nuevo planteamiento de las demandas de movilidad, uno de los medios que se están mostrando más efectivos para aliviar este problema es la aplicación de tecnologías telemáticas en las redes de transporte de superficie. El abanico de avances y soluciones tecnológicas aplicados en las redes de transporte de superficie que se emplean para reunir, procesar y difundir información y para gestionar el tráfico, concebido como un conjunto de elementos interrelacionados se conoce como Sistemas de Transporte Inteligentes (ITS por sus siglas en inglés), aunque anteriormente ha recibido otras denominaciones tales como Telemática Avanzada del Transporte (ATT, *Advanced Transport Telematics*) e Informática aplicada al Transporte por Carretera (RTI, *Road Transport Informatics*).

Esta tecnología, concebida como un todo con identidad propia (aunque lógicamente relacionada con áreas similares) se basa en tres pilares cual son

la aplicación de los computadores,

los sistemas de información y

la tecnología de las comunicaciones,

siendo su objetivo fundamental mejorar el movimiento de personas y mercancías en el transporte por carretera (y sus relaciones con otros medios de transporte) en sus tres dimensiones: incrementar la eficiencia del sistema, mejorar la seguridad vial y decrementar los efectos medioambientales.

El presente capítulo tratará de proporcionar una visión general de los ITS y de las tecnologías de las comunicaciones empleadas en los mismos, con apartados específicos acerca de los trabajos tendentes a la normalización para estos sistemas.

2.1. FUNCIONES DE LOS ITS Y OPCIONES TECNOLÓGICAS

Las aplicaciones de la telemática en el transporte, se pueden clasificar en siete áreas funcionales, según [Nij96], que son:

- sistemas de información
- gestión del transporte público
- gestión de flotas
- gestión del tráfico
- gestión de la demanda
- gestión de aparcamientos
- asistencia a la conducción

Esta clasificación es un tanto ambigua, dado que algunas funciones se relacionan estrechamente con más de un área. La dinámica de la evolución de esta tecnología no aconseja encorsetar las funciones en compartimentos estancos, pero esta clasificación nos resultará conveniente para una exposición introductoria.

2.1.1. Sistemas de información al usuario

La provisión de información del viaje y del tráfico puede considerarse la base de la telemática del transporte, estando relacionada con todas las demás áreas, que necesitan esta información. Es más, el flujo de esta información constituye un flujo paralelo al flujo del tráfico que, por ser más rápido que éste, debe contribuir a la mejora de la eficiencia del mismo.

Los sistemas avanzados de información al usuario (ATIS, *Advanced Transport Information Systems*) constituyen una subárea muy importante de los sistemas de información en transporte, y se basan en la premisa de que cuanta más información se proporcione al viajero sobre el trayecto y sobre las condiciones del tráfico, este podrá utilizarla en su beneficio con una mejora global de la eficiencia del sistema. Las formas más simples de los ATIS son los informes de tráfico por radio comercial, las emisoras dedicadas de información sobre tráfico (HAR *Highway Advisory Radio*) y cualquier otro

medio de difusión de información (incluida la Internet). Los Paneles de Mensajes Variables pueden ser considerados como medio de información, aunque por su capacidad de visualizar señales con iguales efectos a la señalización fija, son mas bien consideradas parte de los sistemas de gestión y control de tráfico. Las tendencias más avanzadas en los ATIS apuntan hacia los sistemas de ayuda a la navegación en el propio vehículo (mediante computador de a bordo), la personalización de la información difundida y el uso de dispositivos de comunicación personal (como telefonía móvil) para transmitir información hacia el viajero, e incluso para que este pueda transmitir información.

Según el tipo de informaciones, pueden dividirse en dos tipos: información sobre el viaje e información sobre el tráfico. La información del viaje incluye información anterior al viaje e información durante el viaje que permite a los usuarios la preparación del mismo. En relación a la información del tráfico, la información de viaje tiene una mayor duración temporal y se refiere a un área geográfica mayor. La información del tráfico para conductores comprende información dinámica sobre el estado del tráfico que es específica de un intervalo temporal y un área geográfica delimitada.

La información a los usuarios antes de emprender un trayecto (sea información de viaje o sea de tráfico) es difundida a través de medios clásicos (prensa, radio, televisión -teletexto-) o mediante interfaces especiales (públicos o privados). En este último caso la información suministrada puede ser estática o interactiva, e igualmente puede obtenerse sin establecer comunicación con fuentes externas, con comunicación unidireccional o mediante una comunicación bidireccional. Por su parte, la información a los conductores durante el trayecto se realiza bien por medios colectivos (Paneles de Mensajes Variables) bien mediante terminales individuales situados en el vehículo para ayuda a la navegación.

La mayoría de estos sistemas de ayuda a la navegación requieren algún tipo de comunicación (obviamente móvil) entre el vehículo y el entorno. Según sea esta comunicación, los sistemas de información al conductor (en el vehículo) se clasifican en cinco categorías [OECD88]:

Sistemas de navegación autónomos, que no requieren comunicación con el exterior. Son sistemas autocontenidos que tratan de determinar por si mismos su situación respecto al entorno.

Sistemas de difusión zonal, con comunicación unidireccional del centro de control al vehículo. Permiten difundir información a un área de decenas o cientos de kilómetros.

Entre las más desarrolladas de las técnicas posibles, se encuentra el RDS (*Radio Data System*) que permite superponer mensajes codificados digitalmente a las emisiones normales de radiodifusión VHF-FM. Según el terminal y el software instalados en el vehículo, estos mensajes tendrán mayor o menor funcionalidad. La utilización de esta técnica para la transmisión de información de tráfico se conoce como RDS-TMC (*Radio Data System - Traffic Message Channel*) y tiene una proyección muy importante en los programas europeos de difusión de información para el transporte.

Sistemas de transmisión local con enlace unidireccional desde dispositivos situados en la propia carretera hacia el vehículo. Tienen el mismo objetivo que los sistemas de difusión zonal pero con una mayor densidad de transmisores siendo los medios más utilizados microondas e infrarrojos, más versátiles para las distancias cortas. Básicamente tienen tres usos: transmisiones de advertencia de peligro, balizas de localización (para determinar la posición) y combinaciones de ambos para guiado del vehículo (el vehículo recibe instrucciones de las balizas que bordean la vía).

Sistemas de comunicaciones móviles, con enlace bidireccional entre un centro de control y el vehículo. Existen varios sistemas de comunicaciones por radio implantados; sin embargo, estos sistemas están limitados en cuanto al número de usuarios que pueden atender. Por ello, este tipo de sistemas tiende hacia el uso de sistemas de comunicación por satélite y de redes públicas de telefonía móvil, especialmente GSM (Sistema Global para Móviles). En los sistemas celulares, el móvil se comunica con el transceptor de la celda en la que se encuentra y a través de este con cualquier nodo accesibles por las redes públicas de telefonía.

A diferencia de los sistemas unidireccionales, el vehículo puede transmitir hacia el centro de control, permitiendo desde llamadas de emergencias a consultas interactivas. Una posibilidad relevante de esta capacidad es la emisión periódica por parte del vehículo de su posición y estado, lo cual es utilizado en aplicaciones como gestión de flotas, y también para la determinación de las condiciones del tráfico y la estimación de los tiempos de viaje.

Sistemas de comunicación local, con enlace bidireccional entre el vehículo y dispositivos situados en la propia carretera. Estos sistemas comunican unidades situadas a pie de vía cercanas a las intersecciones con unidades especiales a bordo del vehículo. Los objetivos de estos sistemas son los mismos que los del caso anterior, pero con una descentralización de los transmisores y un especial énfasis en los sistemas de autonavegación y autoguiado (los sistemas de autoguiado también se llaman AVCS - *Advanced Vehicle Control Systems*).

Además del tipo de comunicación los sistemas de información al conductor también se caracterizan por la interfaz que ofrecen. Básicamente hay tres tipos de interfaces: ayuda direccional que proporciona una sencilla información de la situación y la dirección a seguir; sistemas de mapas que con ayuda de dispositivos de almacenamiento de alta capacidad permiten mostrar mapas sobre la posición y el trayecto; y sistemas de guiado que además de las funciones anteriores utiliza información y algoritmos para identificar la ruta más adecuada para llegar al destino.

2.1.2. Gestión del Transporte Público

Las aplicaciones relacionadas con la Gestión del Transporte Público son también referidas como APTS (*Advanced Public Transport Systems*). Se trata de aplicaciones para mejorar la eficiencia y la calidad del servicio a los usuarios del transporte público: sistemas de información, sistemas de pago automático y sistemas de localización para mejorar la gestión de la flota e informar a los usuarios.

En este campo, se pueden distinguir cinco tecnologías, algunas de las cuales se solapan con aspectos de los ATIS:

Sistemas de Monitorización y Localización Automática de Vehículos (AVM y AVL) utilizan las comunicaciones para conocer en un centro de gestión el estado y situación de los vehículos. Se utilizan multitud de técnicas y no hay una técnica dominante debido a la falta de armonización de los desarrollos privados en este ámbito.

Terminales interactivos de información a los viajeros, de iguales características a los sistemas de información previa al viaje, suelen ofrecer información estática de muy diversa índole (trayectos, horarios, precios, ...) y utilizan escasamente la comunicación de datos.

Sistemas de información dinámica (tiempo real) al viajero, se diferencian de los anteriores en que suministran información dinámica, por tanto constantemente actualizada a través de un enlace de comunicación, de los tiempos de salida y de llegada de los diversos vehículos.

Sistemas de pago automático, con diversos desarrollos que incluyen entre otros máquinas expendedoras, dispositivos de validación y cancelación y tarjetas inteligentes.

Estos dispositivos utilizan las comunicaciones por ejemplo para realizar reservas o para la transferencia de dinero electrónico.

Sistemas de priorización del transporte público: para dar prioridad al transporte público se dispone de opciones como los carriles dedicados, los sistemas de '*gating*' y el tratamiento priorizado en los tramos señalizados. Los sistemas de control de tráfico urbano (tratados en otro subapartado) encuentran dificultades para compatibilizar la optimización del flujo de tráfico con la prioridad para el transporte público; este aspecto forma parte de los desarrollos actuales en sistemas integrados de gestión del tráfico urbano.

2.1.3. Gestión de Flotas

principalmente concernientes a la gestión de mercancías y flotas, comúnmente se denominan sistemas FFM (*Freight and Fleet Management*; también se conocen como aplicaciones CVO - *Comercial Vehicle Operation*), pretenden mejorar la eficiencia de las compañías de transporte por medio del intercambio electrónico en tiempo real de datos e informaciones, principalmente para la gestión, planificación y monitorización del transporte de mercancías.

En la actualidad se distinguen cuatro tecnologías principales que se aplican a la gestión de flotas: intercambio electrónico de datos; identificación automática de cargas, vehículos y conductores; localización automática y comunicaciones bidireccionales y sistemas de navegación. En este ámbito, los sistemas, equipamiento y software que se utilizarán en la próxima década ya están disponibles comercialmente o como prototipos, incluyendo equipos de a bordo, programas especializados, equipos automáticos o semiautomáticos de captación de datos de las mercancías e infraestructura de telecomunicaciones (principalmente por satélite y por sistemas celulares).

Intercambio electrónico de datos. El término EDI (*Electronic Data Interchange*) define el intercambio electrónico de documentos y órdenes referentes al transporte entre transportistas, clientes y destinatarios en un formato normalizado. Independientemente de la infraestructura de comunicaciones, la barrera más importante para dicho intercambio de datos ha sido tradicionalmente la falta de estándares en cuanto a los formatos, principalmente motivada por la existencia de desarrollos propietarios que dificultan o impiden la comunicación entre sistemas de información diferentes.

La base del intercambio efectivo es la adopción, dentro de un modelo de referencia común (el modelo de referencia OSI que se comenta más adelante), de un acuerdo sobre la estructura de los mensajes, la codificación y los procesos involucrados en las relaciones comerciales del transporte de mercancías para una exacta interpretación de la información transferida entre los participantes. Esto requiere un esfuerzo de todos los participantes los diferentes sistemas de cada uno de ellos se integren a nivel lógico mediante el EDI.

Identificación automática de cargas, vehículos y conductores. El objetivo de la identificación automática es obtener la información digital necesaria sin procesamiento humano. Las tecnologías de identificación automática más difundidas son los códigos de barras, las bandas magnéticas, las tarjetas inteligentes, siendo los códigos de barras los de más amplia aceptación. Otras tecnologías que se avecinan para esta función son el reconocimiento de caracteres, el reconocimiento de voz y la identificación por radiofrecuencias.

Todos estos sistemas comprenden un conjunto de elementos: un código de identificación del objeto, una representación de dicho código, un soporte para dicha representación, un equipo para incorporar la representación en el soporte, un lector de la representación contenida en el soporte y un descodificador para convertir la representación a un formato digital y transferirlo a un computador.

Los avances en la identificación por radiofrecuencias eliminan uno de los principales inconvenientes de los sistemas de identificación automática cual era la necesidad de un contacto o de una proximidad 'estática' entre lector y objeto que debe identificarse, permitiendo la identificación automática incluso de móviles a gran velocidad y abriendo un amplio abanico de posibilidades: peaje sin detenciones, identificación de camiones y guiado en grandes superficies, identificación de contenedores en transporte intermodal, reducción de retrasos en puntos de parada obligatoria (p.e. fronteras) mediante la identificación anticipada de los vehículos, identificación de conductores no autorizados por la combinación de las tarjetas inteligentes de identificación con las tecnologías de radiofrecuencia, etc.

Localización automática y comunicaciones bidireccionales. Los sistemas de localización automática permiten a los operadores de flotas monitorizar el movimiento de los vehículos. Los medios utilizados permiten asimismo la comunicación bidireccional con el conductor. Estos sistemas ya están disponibles comercialmente, dotando a los vehículos de un equipo de a bordo y a los centros de operación de un enlace de comunicación y un software especializado para la monitorización y gestión de la flota.

Las opciones técnicas abarcan desde el uso de tecnologías basadas en redes públicas a la utilización de sistemas de radio privados (sistemas de *trunking*). El problema principal de éstos es la limitación en la cobertura del sistema y en el número de usuarios. La utilización de redes públicas, sean sistemas de comunicación por satélite o sistemas celulares permiten coberturas mucho mayores.

Es especialmente relevante a estos fines el desarrollo de la tecnología GSM (Sistema Global para telecomunicaciones Móviles) un sistema de tecnología digital con las mismas posibilidades que la comunicación por satélite pero mucho más económico que además ha sido adoptado por la mayoría de los países europeos convirtiéndose en un estándar de facto.

Sistemas de navegación. Los retrasos debidos a problemas del tráfico son especialmente onerosos para el sector del transporte de mercancías. El efecto de las congestiones sobre los camiones es desproporcionadamente mayor que sobre los vehículos privados. Por ello, la ventaja económica de los sistemas de información dinámica es mayor para los conductores de camiones que para los de coches, reduciendo los costes debidos a retrasos y a periodos sin carga.

Los sistemas de navegación y guiado son aplicaciones telemáticas que permiten al conductor seguir una ruta óptima hacia su destino. Básicamente se trata de las mismas aplicaciones de ayuda a la navegación para todo tipo de vehículos, ya comentadas en un punto anterior, pero teniendo en cuenta que las distancias recorridas y por tanto el área geográfica a cubrir son por lo general mayores que en los desplazamientos privados, con el consiguiente efecto sobre los sistemas de comunicación e información requeridos.

2.1.4. Gestión del Tráfico

La gestión del tráfico y de la red viaria es una tarea de organismos gubernamentales de diversos ámbitos y de los responsables y operadores de las infraestructuras. Los Sistemas Avanzados de Gestión de Tráfico (ATMS por sus siglas inglesas) comprenden un amplio grupo de aplicaciones que tratan la monitorización en tiempo real de las condiciones del tráfico y de las vías, la captación de datos, el procesamiento de los mismos y las medidas de control de tráfico, basadas en los datos monitorizados (incluyendo previsiones) [Obi96].

Las diferentes funciones de este campo precisan los medios adecuados para la captación de datos, transmisión y técnicas de procesamiento para el soporte de las decisiones de gestión y para proporcionar información actualizada a los conductores del tráfico y del estado de las vías. Ejemplo de estos medios son los sensores para la captación automática

de datos del tráfico (flujo, velocidad y ocupación), sensores de niebla y de hielo, sistemas de control de puentes y túneles, señalización variable (incluyendo paneles de mensajes variables) y postes de auxilio en carretera. Estos medios pueden ser empleados para permitir la ayuda a la decisión, actuando sobre el tráfico (p.e. modificando los ciclos y los repartos de un grupo semafórico).

Se trata de un sistema de control, con la sensorización, el procesamiento y la actuación. Tradicionalmente, el trabajo desarrollado en Ingeniería de Tráfico se refiere a este lazo cerrado sensorización-procesamiento-actuación-sensorización para comprobar la efectividad de la actuación. Uno o más de estos procesos aún son frecuentemente realizados por seres humanos (adquisición de datos por observación directa o a través de cámaras de televisión; procesamiento y toma de decisiones a cargo de operadores; actuación por medio de agentes en la vía pública). A medida que se avance en los sistemas de computación y de comunicación, aumentará el grado de automatización de estos procesos. (Cuánto contribuye esta automatización a la mejora del tráfico en sí queda fuera del alcance de este estudio; se trata aquí de determinar si determinadas estrategias y modelos de gestión automatizados pueden ser llevados a la práctica).

El área funcional de la gestión y control del tráfico se desarrolló anteriormente a todas las demás, existiendo muchos de éstos sistemas instalados[1] tanto para la gestión del tráfico urbano como para el interurbano. Algunos de los subsistemas y sus opciones tecnológicas en este área se describen a continuación.

Sistemas de monitorización. Los sistemas de monitorización incluyen sistemas de sensorización como detectores de lazo inductivo (herederos de los tubos neumáticos aún en uso), circuito cerrado de televisión, equipos con cámaras y procesamiento local de la imagen (conocidos como equipos DAI, Detección Automática de Incidentes [Mar95]). Los detectores de lazo inductivo son los más difundidos al presente; los sistemas de videovigilancia están entrando en una nueva generación con los citados equipos DAI. Los

[1]El caracter precursor de los sistemas de control de tráfico genera dos problemas que deben enfrentarse: en primer lugar, los primeros sistemas desarrollados de manera independiente para cubrir necesidades concretas utilizan equipamiento y soluciones propietarias, con escasas posibilidades de integración en sistemas de mayor rango; el segundo es el hecho de que los sistemas que se instalan en un área con sistemas instalados anteriormente, aun incorporando las nuevas tendencias tecnológicas, tienden a heredar la arquitectura de los sistemas precedentes y en consecuencia padecen los defectos indicados para ellos.

datos captados automáticamente, unidos a los informes convencionales de patrullas y teléfonos de emergencia, se muestran muy efectivos en la detección de congestiones e incidentes en zonas de hasta unos cientos de metros. Las variantes más avanzadas de esta tecnología son los sensores de ultrasonidos, operativos únicamente en Japón.

Las previsiones a medio o largo plazo en este campo apuntan a que los datos básicos necesarios para la gestión del tráfico serán obtenidos por medio de comunicaciones entre los propios vehículos y los centros de control de tráfico.

Señalización variable. La señalización variable sufre una fuerte expansión en la actualidad, especialmente en Europa, al objeto proporcionar a los conductores varias clases de información dinámica como obras, incidentes, condiciones meteorológicas, ambientales y condiciones del tráfico y controlar el mismo mediante señales de obligación. De estas últimas, las más frecuentes son las restricciones de uso de determinados carriles y las de control de velocidad (según la convención de Viena de 1968, la señalización variable tiene el mismo carácter regulador que la señalización fija si cumple una serie de normas internacionales de señalización vial [Aur96]).

Las técnicas para visualización de signos en señalización variable abarcan desde viejos sistemas mecánicos hasta más las avanzadas tecnologías LCD y LED. Los contenidos que muestran varían desde simples signos (pictogramas) a mensajes más complejos (textos explicativos de las circunstancias del tráfico o combinaciones de pictogramas y textos). Los sistemas de señalización variable conectados con sistemas de monitorización son capaces de proporcionar la información más exacta y fiable posible.

***Ramp metering* (control de accesos).** Consiste en la incorporación gradual de un flujo de tráfico en una vía congestionada controlando el acceso de cada vehículo por medio de un semáforo o señalización luminosa variable. La incorporación controlada busca cubrir la capacidad disponible en la vía congestionada ajustando al máximo la demanda de acceso, evitando simultáneamente que la superación de la capacidad haga caer la eficiencia de la vía principal y que la limitación gradual del acceso retrase la propagación de la congestión a las vías confluentes a la arteria principal.

Teléfonos de emergencia. En la mayoría de las autopistas y autovías se dispone de una red de teléfonos de emergencia o Postes de Auxilio en Carretera, para ayudar a los conductores en caso de accidentes o problemas técnicos. También pueden servir para informar a las autoridades de incidentes que afectan al tráfico. Por su densidad y su distribución regular (uno cada dos kilómetros en la práctica totalidad de las autopistas: uno

cada 200 metros en los túneles [SET93]) constituyen el equipamiento más frecuente en las autopistas.

Carriles con prioridad. Los carriles con prioridad son sistemas que permiten a determinados grupos de usuarios el acceso prioritario y/o exclusivo a una parte estratégica de la red viaria, según políticas promovidas por los organismos responsables de la gestión del tráfico. Las posibilidades incluyen el tratamiento prioritario de los transportes colectivos en los semáforos o vías de uso exclusivo para autobuses para agilizar el movimiento del transporte público y promover su uso.

Otros ejemplos son los carriles para Vehículos de Alta Ocupación (VAO) también llamados Carriles para Auto Compartido, carriles especiales de acceso exclusivo para vehículos ocupados por más de una persona (a menudo carriles compartidos para transporte colectivo y vehículos de alta ocupación: son los llamados carriles BUS-VAO [Fer96]), y los carriles para vehículos pesados.

2.1.5. Gestión de la demanda

Ninguna de las diferentes formas de control de tráfico solventan los problemas suscitados cuando la demanda supera significativamente la capacidad de la vía. En los casos de saturación, las medidas para restringir la demanda de tráfico son rigurosas pero necesarias para aliviar los problemas del tráfico sin ampliar las infraestructuras. Las aplicaciones telemáticas tienen un papel importante en este intento de ajustar la demanda, que puede hacerse limitando el tráfico a ciertos periodos del día o estableciendo peajes para restringir el uso de determinados segmentos de la red viaria. A largo plazo, la gestión de la demanda del tráfico debe ir ligada al uso del terreno y a las políticas de urbanización para minimizar la longitud de los desplazamientos y la dependencia del uso de los vehículos privados, estableciendo estrategias integradas del transporte y la movilidad.

Los desarrollos telemáticos para el pago de peajes y el control de la congestión avanzan mucho más rápidamente que las decisiones políticas y la aceptación ciudadana de dichas medidas. Contrasta con ello el alto grado de aceptación del pago por aparcamientos y por el uso de transporte público. La medida de gestión de tráfico más común es la restricción de tráfico rodado en los centros urbanos.

El desarrollo tecnológico más importante en éste área es al Identificación Automática de Vehículos (AVI, en inglés) y las tecnologías de peaje automático, ambas irrumpiendo actualmente en el mercado. Incluyen *tags* dispositivos en el vehículo que pueden

comunicarse con dispositivos situados en la vía para controlar las restricciones de acceso (a residentes, disminuidos, o servicios públicos) o permitir el pago de tasas sin necesidad de detenerse. La primera generación de *tags* AVI requiere antenas terrestres y el paso de los vehículos a poca velocidad; la segunda generación emplea enlaces por microondas de corto alcance que permiten la comunicación bidireccional a velocidades medias y altas (el pago de peajes combina la tecnología de comunicaciones de corto alcance con el uso de tarjetas inteligentes [Gem93]).

2.1.6. Gestión de aparcamiento

La gestión de aparcamiento es un elemento que se relaciona tanto con la gestión de la demanda como con el control del tráfico y la información al usuario. Las aplicaciones telemáticas pueden proporcionar soluciones para incrementar la eficiencia de los sistemas de aparcamiento en áreas urbanas mediante la monitorización en tiempo real de las capacidades de aparcamiento, la previsión de plazas disponibles y la difusión de información a los conductores sobre las disponibilidades de aparcamiento.

Dentro de los aparcamientos, los sistemas de guiado hacia las plazas disponibles minimizan el tiempo empleado por los conductores. Funciones adicionales incluyen la reserva y el pago automático del aparcamiento y los sistemas automatizados sancionadores de aparcamientos incorrectos. Las tecnologías empleadas para estas funciones son similares a las empleadas en las áreas de gestión de la demanda, gestión del tráfico y información al usuario.

2.1.7. Asistencia a la Conducción

La asistencia a la conducción engloba la monitorización de los conductores, los vehículos y sus alrededores, proporcionando asistencia directa o indirecta a la conducción por medio de estímulos dirigidos al conductor o por el control directo del vehículo, aunque esta última [Pov96] es más bien una opción a largo plazo, pues implica una organización completamente nueva de los vehículos y sus infraestructuras que constituiría en sí un nuevo modo de transporte.

Las tecnologías claves en este campo, disponibles de serie u opcionalmente en vehículos privados y comerciales , son:

Sistemas de radar para automóviles. La tecnología radar es la opción principal para la monitorización continua de la distancia y velocidad de y entre vehículos. Estos sistemas consisten en un transmisor fijo en el vehículo que emite señales que se reflejan en los objetos colindantes. Las señales reflejadas son captadas y procesadas en combinación con información de la velocidad y dirección del propio vehículo, determinando en su caso la existencia de situaciones peligrosas. En tal caso, el conductor recibe una señal luminosa y/o acústica. Estos sistemas pueden ir combinados con sistemas de control de la velocidad y con sistemas de frenado anitbloqueo (ABS).

Sistemas de monitorización del estado del conductor. Estos sistemas detectan cambios significativos en el comportamiento del conductor mediante la monitorización continua de la magnitud y frecuencia de las desviaciones del comportamiento del conductor respecto a su comportamiento habitual. Éste es establecido mediante mediciones fisiológicas como electroencefalogramas, electrooculogramas y videovigilancia del rostro del conductor (en particular de las pupilas) o medición de parámetros relativos al vehículo, como el ángulo de conducción, velocidad, aceleración, fuerza aplicada sobre los pedales o temperatura interior del vehículo. Estos parámetros son interpretados y analizados por una red neuronal adaptativa que convierte las desviaciones relevantes en advertencias al conductor o en acciones de control del vehículo.

Módulos de instrucción y aprendizaje. Los módulos de instrucción y aprendizaje pretenden mejorar el comportamiento de los conductores mediante la evaluación del comportamiento de los mismos teniendo en cuenta la experiencia y acciones anteriores de cada conductor. Estos sistemas, en combinación con otros subsistemas de asistencia a la conducción, determinan cuando un conductor precisa realimentación de sus acciones, la naturaleza de esta realimentación y la forma de presentarla.

Sistemas de gestión del diálogo. Los sistemas de gestión del diálogo proporcionan interfaces hombre-máquina mejoradas para la comunicación entre el conductor y los diversos sistemas telemáticos. Por medio de algoritmos de control de diálogo y estimaciones de la carga de trabajo del conductor, los sistemas adaptan la presentación de mensajes al conductor según la capacidad de atención de éste.

2.2. ARQUITECTURA DE LOS ITS

A falta de una coordinación adecuada, cada empresa o grupo desarrollador de ITS corre el riesgo de diseñar un sistema "ad hoc" para dar soporte a cada servicio, siendo posible (incluso frecuente) que un mismo servicio sea soportado por arquitecturas diferentes. Este hecho, en principio no tenido en cuenta en la evolución de los sistemas ITS, dificulta enormemente las relaciones entre los diferentes servicios o entre servicios similares y coexistentes basados en arquitecturas diferentes (por ejemplo, la relación entre centros de control de tráfico de ciudades adyacentes, o la relación entre el centro de control de una ciudad y el centro de control de las carreteras que confluyen en dicha ciudad [Wer96]).

Para evitar este problema, los servicios derivados de la utilización de la tecnología de los ITS deberían basarse en una arquitectura apropiada, que establezca consideraciones referentes a los tres pilares básicos de los ITS ya citados: aplicación de las computadores, sistemas de información y tecnología de las comunicaciones.

Lo anterior no significa necesariamente que todos los servicios deban basarse en una misma organización, sino que un modelo de arquitectura global común puede contribuir a facilitar la interrelación entre servicios, interrelación ésta que debe considerarse básica desde una concepción global de los ITS como una nueva tecnología con entidad propia (y no como una amalgama de otras muchas). Desafortunadamente, el avance hacia esta interrelación progresa muy lentamente, a pesar del incremento en la demanda del mercado que sin embargo no considera este tipo de relación en sus sucesivas instalaciones, especialmente en Europa, donde es más frecuente que los esfuerzos comerciales se concentren en los sistemas por separado sin mantener una visión global.

2.2.1. ¿QUÉ ES UNA ARQUITECTURA DE ITS?

Una arquitectura de ITS es un marco general para el diseño e implementación de ITS. Esto supone:

a) Definir los servicios que van a proporcionar.

b) Especificar las funciones necesarias para proporcionar estos servicios. Así por ejemplo, la función "detección de incidentes" es necesaria para el servicio "respuesta a estados de emergencia en la carretera".

c) Definir los subsistemas físicos y los requisitos de comunicaciones entre los mismos.

d) Definir las interfaces y los flujos de información entre subsistemas, con una definición de las interfaces considerados como claves y de los atributos que se espera que éstos posean.

Una arquitectura de ITS también supone una demostración de cómo la actuación de los diversos actores implicados pueden integrarse para trabajar de forma efectiva mediante la incorporación de un marco estratégico. Éste no tiene que llegar a ser una especificación o un diseño detallado, ya que cada proyecto de implementación puede cambiar a lo largo del tiempo. No obstante, las funciones requeridas sí deben mantenerse de forma constante.

Éste marco estratégico juega el papel de un plan de viabilidad que indica cuáles son los caminos más adecuados "a priori" sin que sea obligatorio seguirlos al pie de la letra. En cualquier caso, debe asegurarse la armonización de servicios y el consenso con la industria. Este consenso implica la inclusión de una serie de hipótesis realistas referidas al presente y al futuro de los ITS, por lo que debe contar con:

- un análisis de las opciones técnicas disponibles
- la validación de los requisitos
- una estimación de los costes globales
- el análisis coste / beneficio
- una relación de todas las restricciones que puedan afectar la operatividad de los ITS

Determinadas organizaciones que agrupan a fabricantes, proveedores de servicios, operadores de autovías, operadores de telecomunicaciones y administraciones públicas tratan de establecer un modelo de arquitectura para los ITS en diversas áreas y con resultados desiguales [NPB96].

2.2.2. SITUACIÓN EN EUROPA

Las actividades de investigación en Europa de las aplicaciones telemáticas en el transporte formaron parte del programa DRIVE para identificar posibles mejoras en la seguridad del tráfico, en la reducción de la congestión y del impacto medioambiental. Estas investigaciones se han concentrado en el desarrollo de aplicaciones y la coordinación de los diferentes actores involucrados en los problemas del transporte. El programa europeo

también ha impulsado esfuerzos para obtener estandarización a nivel europeo mediante participación o relación con el CEN (Comité Europeo de Normalización), CENELEC (Comité Europeo de Normalización Electrotécnica) y ETSI (Instituto Europeo para la Estandarización de las Telecomunicaciones) [EC96].

Simultáneamente, los proyectos de coordinación técnica CORD/CORDEX [Pat94] han perseguido facilitar el desarrollo de acuerdos en el establecimiento de líneas maestras para la valoración de aplicaciones telemáticas en el transporte. Estos proyectos han sido fundamentales en el establecimiento de unas especificaciones comunes y han sido coordinados por ERTICO (*European Road Transport Implementation Coordination Organisation*), una organización de empresas y administraciones similar a otras que han surgido en Estados Unidos (ITS America) o Japón (VERTIS).

El esfuerzo europeo ha producido aplicaciones punteras en los ITS Sin embargo los intentos de coordinación, a finales de 1996, no han conseguido establecer una arquitectura común, habiéndose limitado a establecer una terminología común y especificaciones básicas para el intercambio de datos y para el desarrollo de aplicaciones específicas. Nuestros encuentros y análisis han demostrado que ello podría ser debido a dos motivos principales:

1) las diferencias en la naturaleza y el tipo de los servicios y las necesidades relacionadas con el transporte por carretera entre los diferentes países y regiones de la Unión Europea, que impiden la compatibilidad y/o la continuidad de los servicios;

2) los intentos de comercializar y estandarizar arquitecturas en competencia y productos incompatibles que crean la incertidumbre en los inversores debilitando la expansión de los ITS

Para asegurar la interoperatividad dentro de Europa todas las partes implicadas deben cooperar en el establecimiento de una arquitectura y en el desarrollo del mercado de ITS pudiendo competir en la provisión de subsistemas y servicios que se adhieran a dicha arquitectura [Pat96].

2.2.2.1. ESPECIFICACIÓN DE SISTEMAS UTMC

El desarrollo más importante en Europa para definir una arquitectura global es la especificación de sistemas UTMC (*Urban Traffic Management and Control*) impulsada por el Departamento de Transporte del Reino Unido. Este desarrollo parte de un estudio

sobre los sistemas de control de tráfico urbano (CTU) existentes llegando al siguiente diagnóstico [RKR96] respecto a los sistemas de CTU existentes:

no permiten la distribución eficiente de información entre sistemas;

no suministran la información en un formato útil para los conductores o viajeros;

no hacen el mejor uso de la tecnología disponible de computación y comunicaciones;

son claramente mejorables en sus capacidades de suministro, expansión y desarrollo.

A partir de este diagnóstico, se propone una arquitectura para sistemas UTMC, que deben ser sistemas modulares, a partir de componentes que pueden ser suministrados por distintos proveedores aunque capaces de funcionar conjuntamente gracias a una definición estricta de las transmisiones de datos entre componentes usando estándares abiertos de uso común (que incluyen un diccionario de datos que define el significado y la estructura de los datos y la especificación de un conjunto muy limitado de protocolos de comunicación). La flexibilidad del concepto UTMC favorece la competencia entre suministradores y el uso de las mejores opciones de comunicación disponibles

Los sistemas UTMC constan de un conjunto de nodos y enlaces de comunicación, definiendo para un sistema UTMC cinco tipos de nodos:

tipo A - sistemas externos: pueden ser otros centros de gestión de tráfico urbano, centros de control de las autovías, centros de emergencias, centros proveedores de información, o otros nodos similares;

tipo B - centro de gestión UTMC: existe un único centro lógico en cada UTMC (aunque físicamente puede estar distribuido);

tipo C - estaciones intermedias (*outstations*): son nodos inteligentes capaces de actuar de forma autónoma;

tipo D - unidades controladas: son unidades 'pasivas' en tanto no actúan de modo autónomo: incluyen Paneles de Mensajes Variables, semáforos, sensores;

tipo E - móviles (equipos de a bordo): desde dispositivos pasivos que son meros receptores a unidades con capacidad de procesamiento.

Los enlaces entre los nodos, que se muestran en la figura, debe cumplir lo siguiente:

las comunicaciones con sistemas externos sólo se producen a través de nodos tipo B;

las estaciones intermedias se comunican entre sí, con los centros de gestión, con unidades controladas;

las unidades controladas sólo se comunican con estaciones intermedias tipo C;

los móviles pueden comunicarse a través de cualquiera de los nodos (directamente con un nodo tipo B, con un nodo tipo C o a través de un nodo tipo D).

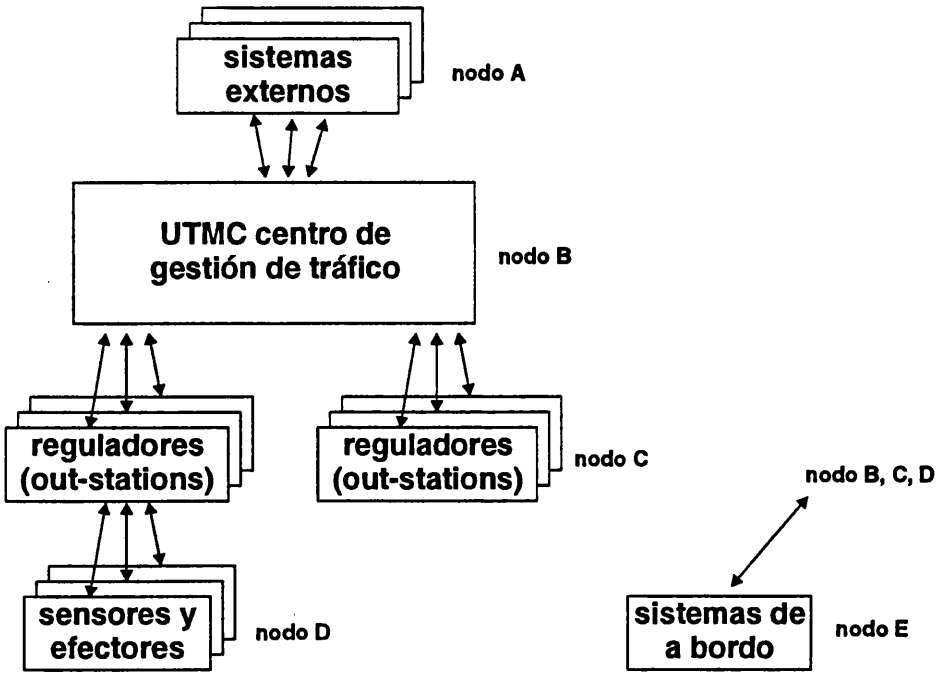


Figura 2.1: Modelo de referencia lógica de los sistemas UTMC

Los servicios de comunicaciones utilizan protocolos Internet o protocolos X.25, que soportan todo tipo de comunicación excepto video en tiempo real. Se definen cuatro clases de comunicación, según las necesidades:

Clase 1: alta seguridad, alta fiabilidad, alta disponibilidad, con mensajes largos;

Clase 2: transmisión tiempo real, intervalos regulares para mensajes cortos frecuentes;

Clase V: comunicaciones especializadas para transmisión de vídeo tiempo real;

Clase N: rutinas de comunicaciones

La especificación de sistemas UTMC deja algunas cuestiones abiertas a la consideración del usuario o del implementador de cada sistema determinado. Las cuestiones más importantes son la selección entre un sistema centralizado y un sistema distribuido (por cuanto afectan a las comunicaciones necesarias) y la tecnología de comunicaciones a

emplear. La especificación UTMC constituye un modelo de referencia que no da respuestas finales a las cuestiones abiertas.

2.2.3. ARQUITECTURA USDOT-FHWA

A pesar del retraso de USA respecto a Europa en el desarrollo e implementación de los ITS (con diferentes nombres antes citados), recientes iniciativas han dado una gran impulso a los ITS en USA que puede invertir esta situación. Una de estas iniciativas es ITS America, una organización que agrupa a diferentes compañías, principalmente estadounidenses, interesadas en la implantación de los ITS. Con el apoyo del Departamento de Transporte de los Estados Unidos y la FHWA (*Federal Highway Administration*), ha desarrollado una Arquitectura Nacional de los ITS consistente en una estructura común para el desarrollo de los Sistemas de Transporte Inteligentes. No se trata de un diseño estricto, ni un concepto del diseño [ITS96], sino un marco de trabajo alrededor del cual se pueden llevar a cabo múltiples aproximaciones, cada una de ellas ajustada a las necesidades individuales del usuario manteniendo al mismo tiempo los beneficios de una arquitectura común [NAR96s].

La arquitectura define las funciones (p.e. difusión de la información o solicitud de una ruta) que deben ejecutarse para implementar un servicio dado (arquitectura lógica), las entidades físicas o subsistemas donde residen esas funciones (arquitectura física), los interfaces y flujos de información entre los subsistemas físicos y los requisitos de comunicación para los flujos de información (p.e. cableado o sin hilos). Además se identifican requisitos para los estándares que deben permitir la interoperatividad a nivel nacional y regional, así como los estándares de producción necesarios para permitir desarrollos económicamente viables. El anexo B de esta memoria recoge una revisión de un grupo de expertos europeos sobre el impacto de la arquitectura USDOT-FHWA en el desarrollo de los ITS en Europa.

2.2.3.1. Arquitectura lógica

La arquitectura lógica presenta una visión funcional de los servicios al usuario de los ITS. Establece las especificaciones de funciones o procesos necesarios para ejecutar un servicio al usuario, y los flujos de datos o informaciones que deben ser intercambiados entre estas funciones. La arquitectura lógica se representa mediante diagramas de flujo de datos, donde se representan las funciones y los flujos de datos entre ellas, o entre funciones

y entidades externas a la arquitectura. Cada función puede ser a su vez desglosada en subfunciones con sus flujos internos en diagramas subsiguientes [NAR96l].

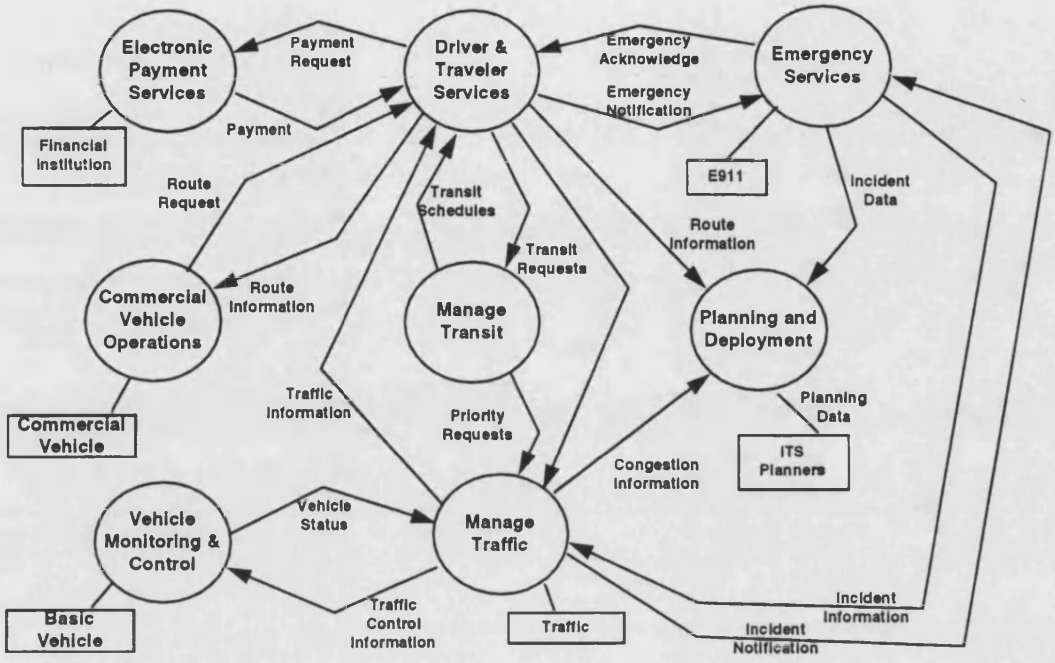


Fig. 2.2: Arquitectura lógica de USDOT-FHWA

2.2.3.2. Arquitectura física

La arquitectura física divide las funciones descritas en la citada arquitectura lógica en sistemas y subsistemas basándose en similaridades funcionales y en el lugar donde se llevan a cabo. La arquitectura física define cuatro sistemas, Sistema del Viajero, Sistema del Centro de Control, Sistema de la Vía y Sistema del Vehículo. Como es habitual, cada uno de los sistemas puede contener diversos subsistemas, y cada uno de los subsistemas se compone de paquetes de equipamiento. Por ejemplo, un sistema de Centro de Control puede contener los subsistemas de Gestión del Tráfico, Gestión de Emergencias y Servidor de Información, pudiendo este subsistema de Gestión del Tráfico estar dotado de una serie de paquetes de equipamiento como paquete de Vigilancia del Tráfico y Control Básico de la Señalización [NAR96p].

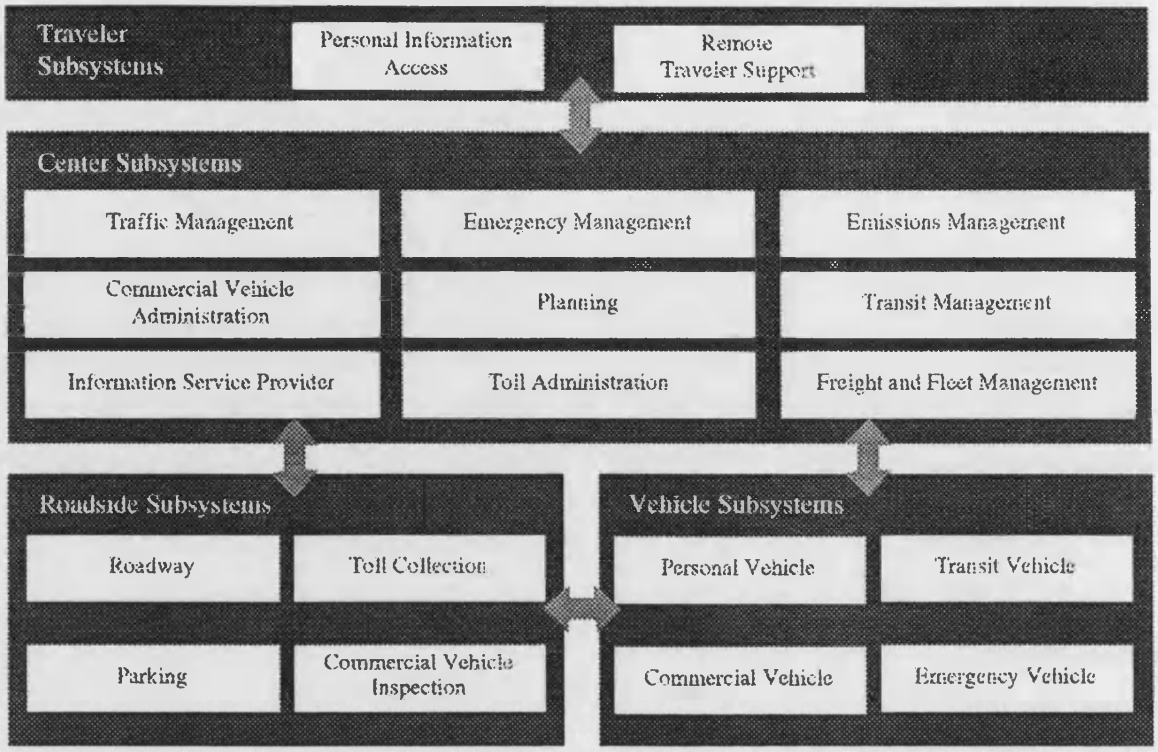


Fig. 2.3 *Diagrama de sistemas en la arquitectura USDOT*

2.2.3.3. Comunicaciones

La arquitectura estadounidense de los ITS proporciona el marco de trabajo para la relación del mundo del transporte con el de las telecomunicaciones para posibilitar el desarrollo e implementación efectiva de los servicios de los ITS. Ante la amplia gama de posibilidades técnicas para la comunicación, la arquitectura identifica y evalúa las diferentes opciones sin seleccionar ni recomendar ninguna de ellas. Uno de los objetivos básicos en el desarrollo de esta arquitectura es permitir la coexistencia entre infraestructuras de comunicación ya existentes con nuevas infraestructuras basadas en las tecnologías más avanzadas, a fin de minimizar el riesgo y el coste de los desarrollos.

La arquitectura americana identifica cuatro tipos de medios de comunicación [NAR96c]:

a) Comunicación por cable (entre puntos fijos): para la comunicación entre sistemas de centro de control, entre estos y los sistemas de la vía, entre sistemas de centro y sistemas del viajero, y para establecer una pasarela con las comunicaciones de largo alcance sin cable.

b) Comunicación de largo alcance sin cable (entre un móvil y un punto fijo): comunicación entre los sistemas del viajero y los sistemas del vehículo.

c) Comunicaciones dedicadas de corto alcance (entre un móvil y un punto fijo): entre los sistemas de la vía y los sistemas del vehículo.

d) Comunicaciones entre vehículos (entre dos móviles): entre sistemas del vehículo.

En la arquitectura del USDOT-FHWA todas las comunicaciones se establecen mediante protocolos estructurados según el modelo de referencia OSI, según se describe en el apartado 2.3.3. de esta memoria.

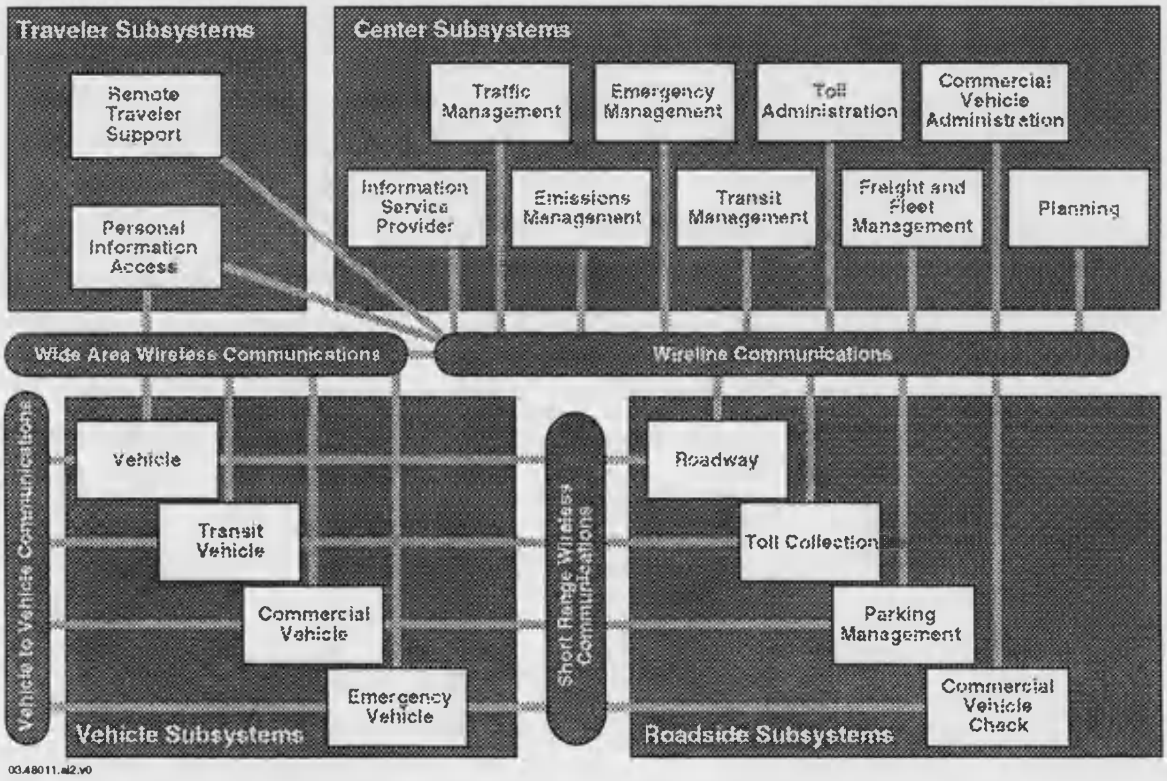


Fig. 2.4 *Diagrama de sistemas y comunicaciones en la arquitectura americana*

2.2.4. ARQUITECTURA PROPUESTA EN ORGANISMOS INTERNACIONALES

El desarrollo de estándares para ITS en Europa se inicia en 1991 con la creación del Comité Técnico TC278 "*Road Transport and Traffic Telematics*" dentro del CEN. Este comité ha desarrollado una serie de documentos para establecer las especificaciones básicas para peaje automático, comunicaciones dedicadas de corto alcance, sistemas de información geográfica y sistemas de información de tráfico a través de RDS-TMC. Simultáneamente, el CENELEC, el ETSI y otros comités técnicos del CEN han desarrollado estándares (algunos en coordinación con el CEN TC278) que cubren radio digital, infrarrojos y microondas para comunicaciones dedicadas de corto alcance, tarjetas legibles por máquinas, códigos de barras, mensajes de intercambio electrónico de datos (EDI) y sistemas de información geográfica.

El progreso europeo en la estandarización motivó la creación dentro de la ISO del TC204 "*Transport Information and Control Systems*" en 1993. Para evitar el solapamiento de actividades y la duplicación de esfuerzos, el CEN y la ISO establecieron un acuerdo según el cual cada organización tendría el liderazgo de un grupo de trabajo.

Como se verá con más extensión en el apartado 2.4. la actividad internacional ha incluido en los distintos organismos internacionales intentos de establecer una arquitectura común. En virtud del citado acuerdo ISO-CEN, el grupo de trabajo 13 del CEN-TC278 "Arquitectura y Terminología" que había experimentado escasos avances, transfirió sus tareas al grupo de trabajo 1 del ISO-TC204 "Arquitectura" cuyas líneas de trabajo son

- Terminología normalizada para modelos de referencia, arquitectura y taxonomía
- Sistemas de Control e Información de Transporte (TICS): términos y definiciones
- Modelo de referencia de Arquitectura

Este grupo trata de definir un número de servicios fundamentales definidos por los diferentes grupos. Sin embargo, la organización de este grupo en dos subgrupos para el desarrollo de la arquitectura, uno para Europa y otro para Norteamérica y el Pacífico, establece un camino tortuoso para el establecimiento de una arquitectura mundial de los ITS al aceptar dicha división de iniciativas.

Se concluye que, aunque los progresos a nivel europeo y mundial para la estandarización de los ITS han producido resultados significativos (se comentan en un apartado posterior), están lejos de establecer una arquitectura de referencia global como la

propuesta por ITS América. Cabe pensar que dicha arquitectura americana tendrá por ello un liderazgo y una importante repercusión en el desarrollo internacional de los ITS.

2.3. TELEMÁTICA DEL TRANSPORTE

Como ha quedado patente en la exposición de funciones, la comunicación de datos ha sido largo tiempo utilizada desde las primeras implementaciones de Sistemas de Transporte Inteligentes , los sistemas de control de tráfico. Sin embargo, las comunicaciones utilizadas en ellos han sido con frecuencia desarrolladas de manera independiente, utilizando medios de comunicación privados y protocolos propietarios.

Por ello, una aplicación en el transporte que emplee la comunicación de datos adoptando soluciones particulares que se desarrollan específicamente para cubrir las necesidades de dicha aplicación de una forma autónoma, pero de modo incompatible con aplicaciones similares, no es una aplicación de la telemática en el transporte.

2.3.1. CONCEPTOS BÁSICOS

Sólo se puede hablar con propiedad de telemática del transporte cuando las comunicaciones utilizadas en los sistemas de transporte inteligentes sigan las directrices siguientes [Ala96]:

- seguimiento de los modelos de organización de las redes de comunicación (particularmente del modelo de referencia OSI) en los subsistemas de comunicación específicos de los ITS;

- utilización de herramientas telemáticas de propósito general para sistemas de comunicaciones (independientes del tipo de aplicación) en el desarrollo, implementación y mantenimiento del sistema de comunicaciones de los ITS;

- adopción de arquitectura de redes generalizadas en la tecnología de comunicaciones para permitir la integración de las redes de comunicación de datos de ITS como subconjunto de las redes telemáticas de uso generalizado, posibilitando de esta manera la comunicación de datos relativos al transporte a través de la infraestructura de comunicaciones disponible para otro tipo de aplicaciones diferentes de los ITS.

Para enmarcar el grado de cumplimiento de este objetivo en los sistemas existentes y en los sistemas planificados en el corto y medio plazo, y a modo de recordatorio, se va a exponer a continuación una visión de la tecnología de las comunicaciones empleada en los ITS

2.3.2. REVISIÓN DE LAS REDES DE COMPUTADORES EN ITS

El concepto de **redes de computadores**, central en la tecnología de las comunicaciones actual, pretende considerar un conjunto interconectado de computadores autónomos, entendiéndose por computadores interconectados aquellos que son capaces de intercambiar información [Bla93]. La idea de interconexión no va asociada a ningún medio físico, ya que coexisten una multiplicidad de medios para la transmisión y recepción de informaciones.

Por otro lado, se considera aquí un concepto amplio del término **computador**, entendido como un sistema basado en microprocesador capaz de funcionar procesando información incluso sin comunicación con el exterior (funcionamiento autónomo) y también capaz de comunicarse con el exterior recibiendo y/o enviando información (sean datos o instrucciones).

Bajo esta perspectiva, la infraestructura de **nodos** y **enlaces de comunicación** que soportan uno o mas servicios de los ITS es una red de computadores, y pueden emplearse en ella la tecnología de comunicaciones de uso común (apuntada en el subapartado anterior). Consideraremos como nodos de la red tanto un sistema de captación de datos de la intensidad de tráfico basado en un microprocesador con capacidades de comunicación como un *mainframe* situado en el centro de control de tráfico de una gran ciudad.

2.3.2.1. TIPO DE ENLACE

La clasificación o caracterización de una red de computadores se puede efectuar atendiendo a distintos criterios, cual son el tipo de enlace, el tipo de conmutación y la extensión geográfica. Atendiendo a la topología de las redes de computadores, también en los ITS se pueden encontrar redes punto a punto y redes multipunto o de difusión.

Las primeras son aquellas donde un canal **punto a punto** 'une' dos computadores permitiéndoles un intercambio directo y privado de datos, aunque no se trate necesariamente de un cable, pudiendo ser también una conexión de microondas, un sistema de transmisión-recepción por láser, o cualquiera de las ya citadas. El diseño basado en canales punto a punto implica que cada nodo en la red está conectado a través de uno o más enlaces punto a punto, a uno o más nodos. La transferencia de información entre dos nodos no contiguos (no conectados directamente a través de un canal punto a punto) implica el paso necesario por estaciones intermedias.

Así como en un sistema basado en canales punto a punto cada uno de los canales es compartido únicamente por dos computadores (uno en cada extremo) las **redes de difusión** se basa en el uso de un único canal compartido por las máquinas que constituyen la red. Las informaciones que un nodo pone en el canal son recibidas físicamente por todos los computadores que lo comparten, aunque únicamente aquél a quien van dirigidos los datos efectúa la recepción lógica de los mismos.

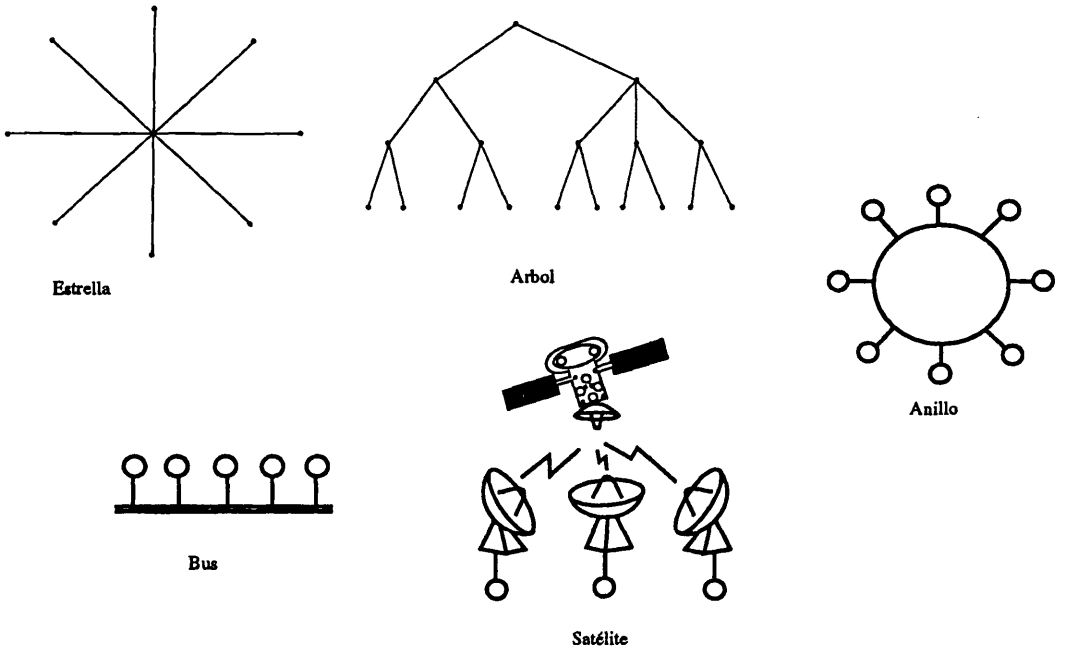


Fig. 2.5: *Topologías de redes punto a punto y de redes de difusión*

Tanto las redes punto a punto como las redes de difusión pueden adoptar diversas disposiciones [Tan91] (como se muestra en la figura 2.5). Las redes empleadas para los

servicios de ITS en entornos urbanos emplean principalmente la topología de enlaces punto a punto.

Por su parte, el uso compartido de las líneas propio de las redes de difusión supone un coste significativamente menor en la implementación de las comunicaciones de los ITS que una multitud de enlaces punto a punto. Por ello, las redes para ITS usadas en entornos interurbanos (donde la estructura geográfica viene a ser longitudinal) o periurbanos son mayoritariamente redes de difusión basadas en una arteria principal de comunicación (de la cual podrán surgir arterias locales). La utilización de líneas compartidas implica unos problemas de capacidad, gestión y fiabilidad que se analizarán en el capítulo tercero.

En las redes de difusión, la gestión de la ocupación del canal compartido se convierte en una función critica. De tal manera que las redes de difusión pueden clasificarse a su vez en [Tan91]:

- Estáticas: a cada estación se le permite el uso del canal durante un cierto periodo de tiempo. La asignación es periódica y cíclica.
- Dinámica: se permite el uso del canal previa petición. La forma de asignación del canal, en este caso, puede ser:
    - Centralizada: un computador hace las labores de árbitro del canal, determinando en todo momento que estación puede transmitir.
    - Distribuida: no existe arbitro, cada estación ejecuta un programa cuyo resultado es la ocupación o no del canal de transmisión.

Nuestro análisis de los sistemas instalados detecta que las redes de difusión empleadas en ITS son siempre centralizadas de manera que en ellas una estación primaria inicia y gestiona todas las comunicaciones dentro la red con las diferentes estaciones secundarias.

2.3.2.2. TIPO DE CONMUTACIÓN

Por último, se puede clasificar una red de computadores de acuerdo a la estructura física que se establece para la transferencia de datos a través de la misma [Bla93]:

- Conmutación de circuitos: aquellas redes donde se establece un circuito físico entre los dos extremos que se comunican. Los medios que componen ese circuito están dedicados durante toda la comunicación, liberándose cuando se produce la desconexión.

- Conmutación de paquetes: la información o mensaje que se intercambian dos computadores se divide en trozos o paquetes antes de ser enviados. Cada paquete puede seguir un camino físico distinto desde su origen hasta su destino. No existen unos medios físicos dedicados a una comunicación en particular, los medios son compartidos por muchas comunicaciones simultáneamente.

- Conmutación de mensajes: similar al caso anterior con la salvedad que se manda el mensaje completo (no fraccionado en paquetes) de una estación de la red a otra.

Obsérvese que la capacidad de que dos computadores pertenecientes a una misma red intercambien información no depende de si la conexión es directa o indirecta. El flujo de información puede fluir directamente entre dos computadores si están ambos conectados por medio de un canal punto a punto (o un canal de difusión) o puede ser encauzado a través de uno o más computadores intermedios para completar el camino del origen a su destino.

El análisis de las redes de comunicación de los ITS no detecta ningún tipo de conmutación preferente: mientras en las redes locales (ver apartado 2.3.5.1.3) instaladas no se utiliza conmutación, en la red superior, que puede ser una red de uso público compartido, se utilizan los tres tipos de conmutación expuestos.

2.3.2.3. EXTENSIÓN GEOGRÁFICA

Atendiendo un criterio según la extensión geográfica, las redes se clasifican en redes de área local, redes de área metropolitana y redes de área extendida.

- LAN (*Local Area Network*)

El concepto de LAN o red de área local en los ITS debe entenderse como una red que ocupa una extensión geográfica delimitada donde todas las comunicaciones se establecen entre nodos que están conectados directamente, de modo que el concepto de área se mantiene aunque pueda corresponder a un área geográfica de extensión muy variable

(desde la instalación para la monitorización y el control de una intersección hasta el conjunto de dispositivos y líneas de comunicación situados a lo largo de un segmento de una autopista).

Las redes de área local empleadas actualmente en los ITS, independientemente de su topología, presentan un nodo llamado regulador de tal manera que las comunicaciones se establecen entre dicho regulador y cada uno de los restantes nodos. El regulador de una red de área local en ITS tiende a tener mayor capacidad de procesamiento que el resto (y por tanto de tomar decisiones) y además es el único con capacidad de comunicarse con nodos ajenos a la red local.

- Las redes para control de tráfico fueron el primer ejemplo de red de área metropolitana MAN, incluso anteriores a las redes telefónicas. Una red de área metropolitana está integrada por varias redes de área local interconectadas (o por un conjunto amplio de nodos que comparten un mismo sistema de comunicación).

Las comunicaciones necesarias para proporcionar un servicio de ITS (por ejemplo, el control de tráfico de una ciudad) se establecen sobre una red de área metropolitana. En los sistemas de transporte inteligente, el tipo de red de tamaño intermedio no está necesariamente restringido al ámbito urbano. Un conjunto de redes de área local que se conectan para soportar un servicio de tráfico interurbano (la gestión de una autopista) constituyen una red de tamaño intermedio que obviamente no se puede llamar de área metropolitana y que llamaremos red de tamaño intermedio.

También constituyen redes de tamaño intermedio los conjuntos de nodos que comparten un mismo sistema de comunicación cubriendo completamente las necesidades de comunicación de un servicio de ITS (por ejemplo la gestión de una flota de camionetas con comunicación mediante *trunking*).

- WAN (*Wide Area Network*)

El concepto de red de área extendida WAN. en ITS supone la relación entre redes de área metropolitana o redes de tamaño intermedio que deben interaccionar, bien sean redes que soportan el mismo servicio en áreas adyacentes, bien sean redes que soportan distintos servicios que deban comunicarse datos (por ejemplo, la red de control de tráfico de una ciudad y la red de difusión de información a los usuarios).

Aunque el objetivo último de los ITS debería ser que todos los servicios ofrecidos estén interrelacionados y sus redes de soporte interconectadas, constituyendo una WAN. de los servicios de ITS, la realidad de los desarrollos tiende a centrarse en cada aplicación

dejando en segundo término la interrelación excepto en casos muy obvios, lo cual no favorece el desarrollo global de los ITS

2.3.3. ARQUITECTURAS DE REDES

La conexión física entre dos computadores no es suficiente para conseguir la transferencia de información entre ellos si no existe un acuerdo sobre cómo debe establecerse la comunicación y cómo debe realizarse dicha transferencia.

Si se define a una **entidad** como cualquier proceso o programa en general que se está ejecutando en un computador, y se denomina como **sistema** a ese computador o hardware sobre el que se ejecuta una entidad. Se puede definir un **Protocolo de Comunicaciones** como el conjunto de normas que regulan la comunicación entre dos entidades de distintos sistemas [Rif92].

Las redes de computadores están organizadas de manera fuertemente estructurada, dividiendo la multitud de procedimientos relacionados con la comunicación de datos entre computadores en módulos separados. Las técnicas de estructuración comúnmente empleadas tratan de cubrir los siguientes objetivos:

- reducir la complejidad lógica del problema en subproblemas más pequeños y en consecuencia más sencillos, dividiendo el protocolo en una serie de **capas** o niveles, separando los distintos procedimientos en esos niveles.

- conseguir simetría para todas las funciones desarrolladas en cada capa para cada nodo perteneciente a la red, esto es, cada nodo tendrá las mismas funciones correspondientes a una capa que los otros nodos de la misma red.

- especificar los interfaces entre las distintas capas o niveles que permitan desarrollar los procedimientos de cada nivel de manera independiente a los de los demás niveles (no es necesario saber cómo resuelve cada capa sus funciones, únicamente es necesario saber cuál es el interfaz que cada capa ofrece a las demás)

- conseguir un lenguaje común para clarificar todos los problemas relativos a la comunicación entre computadores y poder ser usado tanto por los fabricantes de hardware, como por los gestores de la red o los mismos usuarios.

En un modelo estructurado por capas, cada capa es un proveedor de **servicios** que contiene una o más funciones de servicio. Dichos servicios únicamente pueden ser solicitados por la capa superior a la que provee el servicio, como muestra la figura 2.6.

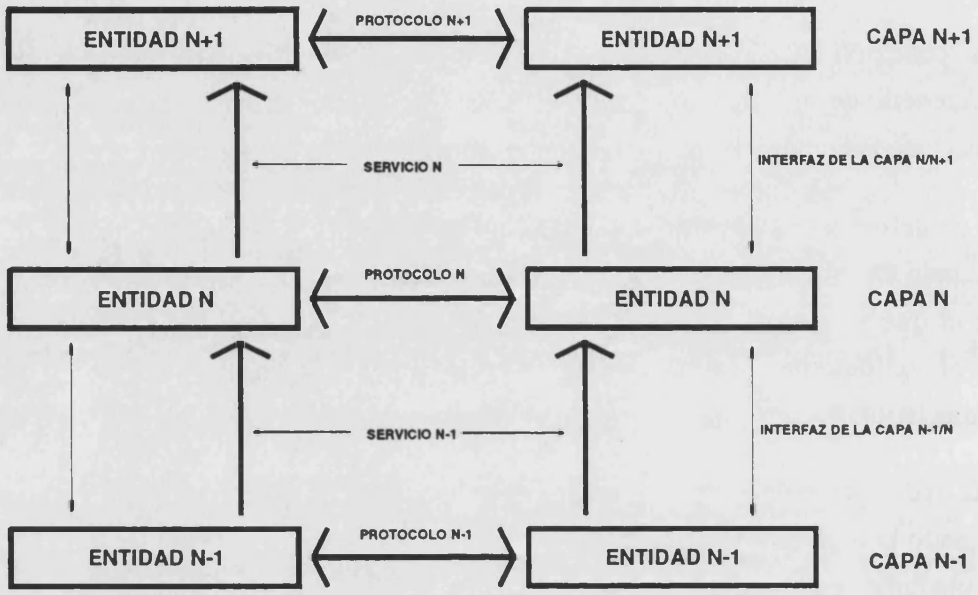


Fig. 2.6: *Organización estratificada y Protocolos de capa*

La capa más baja de la estructura, normalmente conocida como **capa física** ya no tiene por debajo ninguna otra capa lógica a la que solicitar servicios. En su lugar la capa física escribe o lee información en el **medio físico**, que es donde se realiza físicamente la comunicación entre los nodos.

El conjunto de capas y protocolos se conoce como **arquitectura de red**. Por encima del medio físico, el número de capas y los protocolos de la misma son variables, lo cual dificulta enormemente el tratamiento de las comunicaciones entre computadores. Uno de los modelos más comúnmente empleados para determinar la arquitectura de una red es el **modelo de referencia OSI** (siglas en inglés de Interconexión de Sistemas Abiertos), modelo básico en el desarrollo de las telecomunicaciones y que es el propuesto para la organización de las comunicaciones en la arquitectura de los ITS [Tan91].

El concepto básico aplicable a la arquitectura de las comunicaciones de los ITS es la utilización del modelo de referencia OSI, y en éste la separación de la arquitectura de red en dos grandes niveles: el nivel del transporte[2] y el nivel de comunicaciones. Estos dos niveles son claramente separables, aunque deben estar debidamente acoplados para proporcionar el servicio de comunicación de datos a los usuarios de aplicaciones ITS.

El nivel del transporte (*transportation*) comprende las capas 5 (sesión), 6 (presentación) y 7 (aplicación) del modelo de referencia OSI (en la mayoría de sistemas ITS las capas 5 y 6 son absorbidas por la capa de aplicación). El nivel de comunicaciones comprende las capas 1 a 4 (física, enlace de datos, red y *transport*) del modelo OSI. Esta diferenciación de niveles se muestra en la figura 2.7.

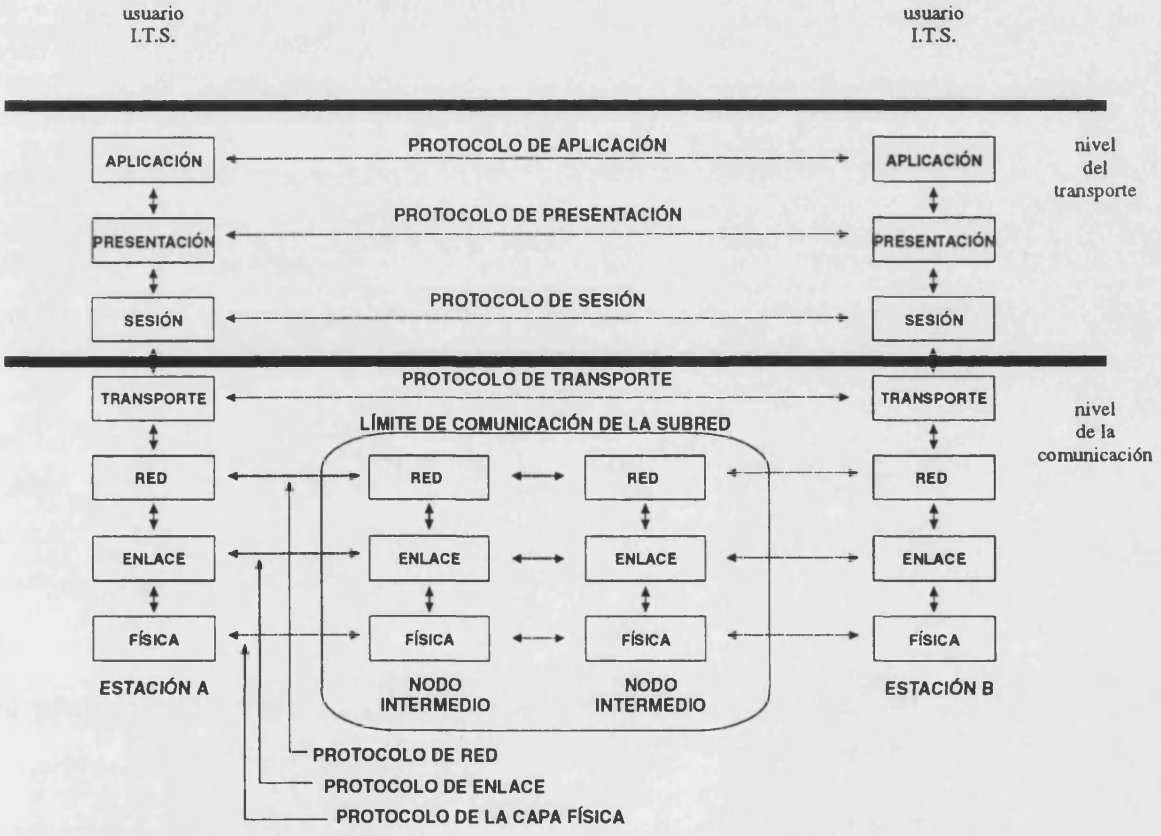


Fig. 2.7: *Niveles de la arquitectura de comunicaciones en ITS*

[2]El nivel del transporte (*transportation*) en la arquitectura de las comunicaciones para I.T.S. se refiere a protocolos relativos a necesidades de los ITS y no al nivel de transporte de datos (*transport*) como cuarta capa del modelo de referencia OSI.

La adopción de este modelo de referencia nos llevará a analizar la ubicación en los distintos niveles de las funciones necesarias. Debe determinarse en cuál de estos niveles se sitúan las funciones de tolerancia a fallos en ITS. Esta memoria mostrará como la ubicación de las funciones de tolerancia a fallos en el nivel de la comunicación, y dentro de éste en la capa de enlace de datos, es la opción más adecuada en un sector de las redes de comunicación de los ITS.

2.3.4. REQUISITOS DE COMUNICACIÓN EN ITS

Los diferentes servicios de ITS tienen unas necesidades desde la perspectiva de comunicación de datos muy diferentes. La arquitectura de comunicaciones de ITS debe tener en cuenta dicha variedad de requisitos.

El nivel de la comunicación tiene dos tipos de componentes: enlaces cableados y enlaces no cableados. Los requisitos de transferencia de datos necesitados por las entidades del nivel del transporte (de ITS) son soportados por uno de éstos o por ambos (la mayoría de los sistemas de comunicación para móviles ligan enlaces no cableados con redes de enlaces cableados).

El nivel de la comunicación se conceptúa como un 'sistema de tuberías' por el cual los datos alcanzan su destino, siendo la mayoría de los detalles de dicho sistema transparentes al nivel del transporte. Las aplicaciones de los ITS, cara al nivel de la comunicación, constituyen una colección de aplicaciones con requisitos varios de transferencia de datos, caracterizados por:

1) los requisitos de distribución y direccionalidad para la transferencia de información (si se trata de comunicación unidireccional o bidireccional);

2) la movilidad de las entidades implicadas (si se trata de una comunicación entre elementos estáticos, entre un móvil y un elemento estático, o entre dos móviles);

3) el volumen de datos a transferir y

4) los requisitos de tiempo y fiabilidad de la transferencia.

La primera caracterización se describe en términos de tipos de servicios: servicios interactivos, que permiten a los usuarios el intercambio de datos en ambas direcciones, y servicios de distribución, que permiten a un usuario enviar un mismo mensaje a un conjunto de usuarios.

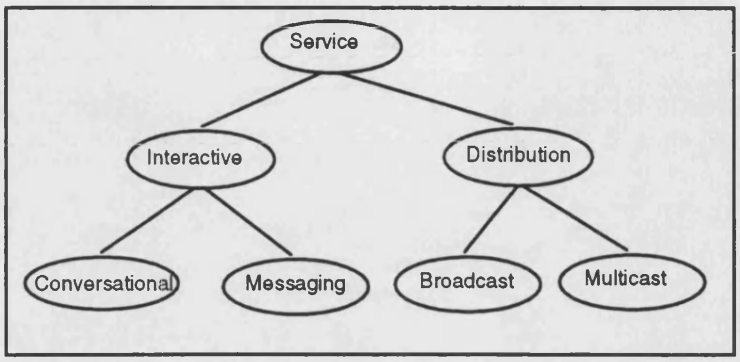


Fig. 2.8: *Tipos de servicios de comunicación*

Una de las líneas maestras en el desarrollo de una arquitectura para los ITS debe ser la posibilidad de coexistencia de infraestructuras existentes con tecnologías emergentes, tanto en los sistemas de transporte como en los sistemas de comunicación, para maximizar la factibilidad de la arquitectura y minimizar el riesgo inherente en la creación e instalación de sistemas, productos y servicios ITS

La definición de la arquitectura para las comunicaciones para ITS permite a los implementadores locales seleccionar tecnologías específicas que, cumpliendo los requisitos de comunicación de los servicios de ITS deseados, mejor se ajusten a las circunstancias específicas de disponibilidades de mercado, restricciones legales, aprovechamiento de infraestructuras existentes, y disponibilidades de financiación, entre otras.

2.3.5. TECNOLOGÍAS APLICABLES EN LOS ITS

En este apartado se describen las tecnologías aplicables para las comunicaciones en los ITS separando el análisis en dos tipos: sistemas de comunicación cableados y sistemas de comunicación no cableados, y estos últimos se dividen en dos clases teniendo en cuenta el alcance de la recolección o diseminación de la información: comunicaciones no cableadas de largo alcance (*wide-area*) o de corto alcance (*short-range*).

2.3.5.1. Sección cableada

Las tecnologías utilizables en la sección cableada constituyen un amplio abanico de posibilidades con coste, capacidad y grado de desarrollo diversos. Con las actuales tecnologías y aplicaciones, la sección cableada de los sistemas de comunicación de ITS no constituye el cuello de botella de los mismos.

Las comunicaciones cableadas permiten el uso de redes públicas, redes privadas o una combinación de ambas. Tecnologías propias de las redes privadas son par trenzado, ethernet, FDDI, SONET, y ATM, entre otras. En las redes públicas compartidas se utilizan principalmente lineas analógicas, *frame relay*, ISDN, ethernet metropolitana e Internet. La opción combinada emplea en lo posible la infraestructura de comunicaciones existente (redes públicas), en ocasiones mejorándola para soportar un mayor volumen de uso, añadiendo los enlaces privados necesarios. Esta opción debe ser la solución preferente para potenciar y rentabilizar la implantación de los ITS

La implementación de los ITS debe seleccionar cualquiera de las tecnologías para redes privadas citadas, siendo todas ellas compatibles con las redes públicas compartidas.

2.3.5.1.1. Tecnologías para redes cableadas privadas

El uso de líneas de cobre (par trenzado) para el nivel más bajo de la red es la opción más económica, y en muchos casos permite la utilización de infraestructura ya instalada. Sin embargo, para sistemas de nueva instalación, el coste de las líneas de fibra óptica no es sensiblemente superior al de las simples líneas de cobre proporcionando prestaciones mucho mayores.

Ethernet es una tecnología de red basada en bus, usada principalmente en redes de área local. El flujo de datos es de 10 Mbps utilizando cable coaxial. El acceso es controlado por un protocolo de acceso al medio con detección de colisiones. Este protocolo por sí solo no puede tratar adecuadamente grandes redes. Para cubrir éstas (ethernet metropolitana) se divide la red en pequeñas redes de área local que se enlazan a través de líneas de alta velocidad. En caso de que se transmiten un conjunto de imágenes de CCTV, el flujo de datos puede superar la capacidad de ethernet y sería necesaria una red separada para la transmisión de las imágenes.

FDDI es una tecnología de red basada en redes de área local usando fibra óptica como medio físico. Soporta un flujo de 100 Mbps y una longitud máxima del cable de 100 Km,

pudiendo conectar hasta 500 estaciones en una misma red. Aunque la topología lógica de la red es un anillo, la topología física puede ser tanto un anillo como una estrella. El acceso al anillo (lógico) se controla por un esquema de paso de testigo: cuando una estación recibe un *token* comprueba si va dirigido a ella y en tal caso lo marca como recibido y lo copia en su cola de recepción; a continuación transmite el *token* hacia la siguiente estación, hasta que llega a la estación que lo originó, que deja de transmitirlo. FDDI II es un estándar mejorado de FDDI con reparto de intervalos temporales que simulan hasta 16 canales a 1144 Mbps más un canal para *token* de 1 Mbps. Este estándar permite transmitir datos a ritmo constante procedentes de cámaras de CCTV a través de algunos de los canales, dedicando los intervalos restantes para la transmisión de datos de los controladores y los sensores.

SONET es un estándar para redes de fibra óptica que permite la interoperatividad entre equipamiento de distinta procedencia. Se define la interfaz física, la velocidad de transmisión, el formato de las tramas, y el protocolo de operación, mantenimiento y supervisión. La frecuencia básica de transmisión es 51,84 Mbps, permitiéndose múltiplos de dicha frecuencia básica. Los datos se transmiten en modo síncrono, a intervalos de 125 microsegundos incluyéndose en cada trama información de supervisión. La información de supervisión permite la monitorización remota de la red para detección de fallos y reconfiguración de circuitos. Las redes SONET se pueden configurar como punto a punto o como anillo. Las redes pueden organizarse como un doble anillo en ambos sentidos para tolerancia a fallos.

ATM es una tecnología de conmutación de paquetes que encamina los paquetes que son multiplexados estadísticamente en un procedimiento *store-and-forward* que utiliza enlaces de diferentes velocidades tratando de maximizar la eficiencia de uso de la red. Utiliza paquetes cortos, llamados celdas, de longitud fija (53 bytes: 48 de información y 5 de cabecera), dando a las celdas diferentes prioridades (p.e. se pueden priorizar las celdas de datos de video sobre otras celdas para favorecer la obtención de imágenes completas). ATM usa un servicio orientado a conexión. Los bytes de supervisión permiten a los nodos de encaminamiento conocer el estado de la red para decidir la ruta y son reescritos por cada nodo. ATM es una tecnología de conmutación de paquetes que puede ser usada sobre diferentes medios de transmisión y favorece el uso de topologías de estrella con líneas dedicadas para cada nodo.

2.3.5.1.2. Tecnologías para redes cableadas públicas

Las tecnologías de redes públicas compartidas disponibles mayoritariamente se muestran en la tabla de la figura 2.9.

Link Technology	Analog leased lines	Digital leased lines	Frame Relay	ISDN
Type of service	Dedicated circuit	Dedicated circuit	Packet switched	Circuit switched and packet
Transmission medium	Standard telephone line	Digital facilities	standard telephone line to four-wire T1 technology	basic rate ISDN - standard telephone lines; primary rate ISDN - four-wire T1 technology
Data rate	up to 28.8	2.4 Kbps, 64 Kbps, fractional T!, T1 (1.5 Mbps), T3 (4.5 Mbps), DS3 (45 Mbps)	56 Kbps up to T1	Circuit switched B channel 64 Kbps, packet D channel 16 Kbps; basic rate ISDN=2B+D, primary rate ISDN = 23B+D
Capabilities	point-to-point and multipoint	point-to-point and multipoint	Suitable for data only.	B channel well suited for CCTV which can be used intermittently, D channel for simultaneous data
Comments	Universally available	High reliability	Fixed monthly charge based on data rate	Cost is usage dependent
Cost/month (rough estimate, based on undiscounted tariffs)		56 Kbps: $300/month; T1: $3.50/month/mile + $2500/month; DS3: $45/mile/month+ $16000/month	56 kbps: $175/month T1: $435/month	basic rate ISDN: $25/month + $0.57/kilopacket for data and $0.016/minute for B channel

Fig 2.9 *Características de algunas redes cableadas públicas.*

Además de las tecnologías listadas en la tabla, algunas ciudades pueden disponer de ethernet metropolitana instalada por las compañías de televisión por cable, aprovechable para la transferencia de datos de ITS Igualmente, la existencia de redes de comunicaciones para el control centralizado de la señalización urbana (principalmente de los grupos semafóricos), normalmente sobre líneas de par trenzado, puede ser aprovechada para el resto de aplicaciones urbanas de los ITS; si se precisa transmitir imágenes de CCTV, será necesaria la instalación de una red adicional de alta velocidad. En tal caso, la red de líneas de par trenzado puede enlazarse, a través de los reguladores o de concentradores, con la red de alta velocidad para incrementar la velocidad extremo a extremo de las transmisiones.

La red Internet, conjunto de redes que utilizan los protocolos TCP/IP, también puede utilizarse como red pública compartida si se resuelven los temas de seguridad y confidencialidad; el uso de Internet debe tomar en consideración la naturaleza estocástica del tráfico en la misma, que produce una varianza en el tiempo de tránsito de los datos y en el porcentaje de paquetes perdidos (hasta un 0.4%). El uso de Internet depende en consecuencia de si las aplicaciones ITS pueden tolerar dicha varianza. Cuando se utilice Internet para las comunicaciones de los ITS se utilizará el protocolo TCP para garantizar la entrega, restringiendo el uso del modo 'datagrama' (UDP) para transacciones no esenciales y/o repetitivas (posición de un vehículo o informes de estado no urgentes).

2.3.5.1.3. Topologías para redes cableadas

La selección de una topología para red cableada debe considerar, entre otros factores, la longitud física total de los enlaces. Los sistemas de control del tráfico, una de las aplicaciones cruciales de los ITS, utilizan actualmente comunicación entre estaciones fijas (sensores y controladores con el centro de gestión del tráfico) que utilizan redes cableadas.

Las redes cableadas que conectan los sensores y controladores con el centro de control se organizan al menos en dos niveles: un primer nivel que conecta los sensores y controladores con un regulador (habitualmente implementado mediante una red privada) y un segundo nivel (que puede utilizar redes privadas o públicas) que conecta los reguladores con el centro de control. Este segundo nivel puede dividirse en niveles adicionales con concentradores o nodos intermedios (centrales de zona), según refleja la figura 2.10.

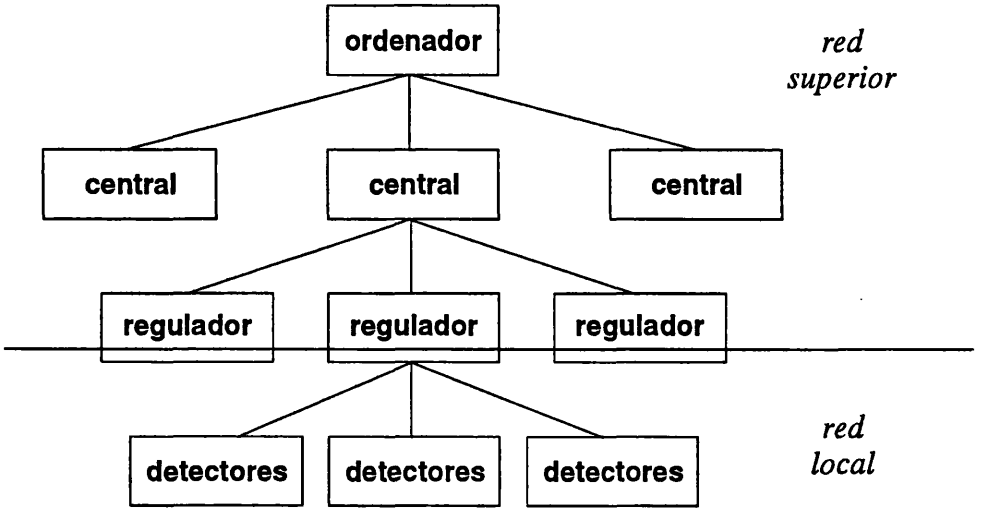


Fig. 2.10: *Jerarquías en redes cableadas para ATMS*

El nivel superior, que comunica los reguladores con el centro de control utiliza topología de estrella sobre enlaces punto a punto de mayor capacidad que los del primer nivel y en ocasiones presenta enlaces entre nodos adyacentes del mismo nivel para permitir el encaminamiento en caso de fallos. La selección de la topología para el nivel inferior depende del medio de transmisión seleccionado, con preferencia por la topología de estrella.

Las aplicaciones soportadas por redes cableadas pueden ser gestionadas de forma centralizada o distribuida, según la capacidad de tomar decisiones de cada uno de los nodos. La gestión centralizada supone que sólo el computador central toma decisiones, y la misión de los nodos intermedios es la de transmitir información hacia o desde dicho computador central. La gestión distribuida consiste en que cada nodo tiene capacidad de tomar decisiones y utiliza la red para comunicárselas a los demás cuando sea necesario o para recibir información de los otros nodos. La tendencia en los ITS es intermedia entre estos dos tipos, dotando a los reguladores de mayor inteligencia y capacidad de tomar decisiones pero supeditadas a la gestión central. El papel de los nodos intermedios (centrales de zona) en los redes de tamaño intermedio de los ITS suele limitarse a implementar la subred para permitir el encaminamiento de los datos entre el computador central y los reguladores.

Sección local de las redes cableadas

Los reguladores se comunican directamente con los nodos de la red local que controlan. Las comunicaciones desde y hacia el resto de nodos de una red local se establece siempre con el regulador. Los reguladores procesan los datos que reciben de los nodos de la red local y envían el resultado del procesamiento hacia el computador central[3]. En sentido contrario, las informaciones u órdenes emitidas desde el computador central son recibidas por el regulador, el cual las procesa, comunicando los datos adecuados a cada nodo de la red de área local.

---

[3]Como ejemplo, los detectores de lazo son interrogados por un controlador local cada 1/240 de segundo, pero la información se comunica al centro de control cada segundo. Igualmente, la salida de un módulo de videodetección debe ser procesada para producir una cuenta de vehículos y/o una medida de la velocidad media, y éstas son las informaciones que se transmiten al centro de control.

El regulador también implementa procesos para monitorizar los dispositivos y las redes de comunicación, para detectar automáticamente los fallos y adoptar medidas reparadoras si fueran posibles y/o necesarias.

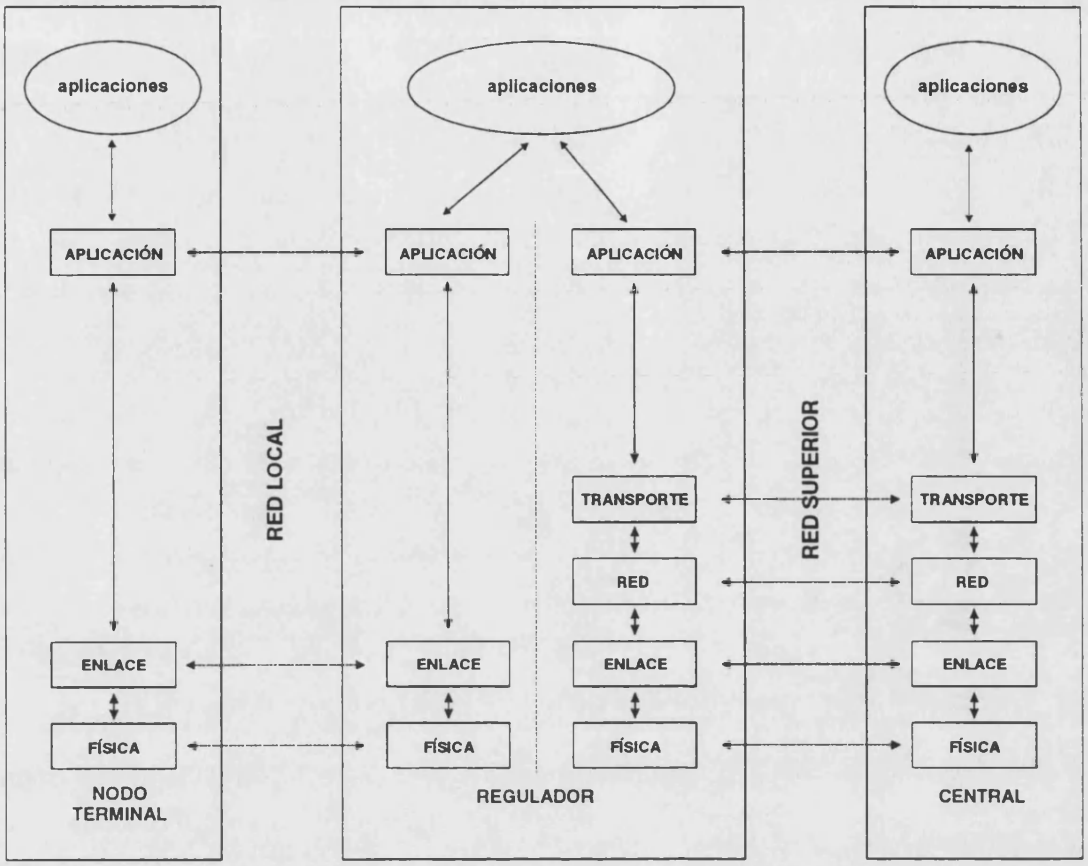


Fig. 2.11: *Diferencia de niveles en una red cableada para ITS*

En esta organización, apreciable en la figura 2.11, la separación de la red cableada en dos tramos supone el uso de protocolos de comunicación diferentes para cada uno: el primer tramo lo constituyen redes de área local sin necesidad de protocolos de encaminamiento; en el segundo, los datos se transmiten a través de redes públicas o privadas con posibilidades de encaminamiento. Esta diferencia de necesidades se analizará posteriormente.

2.3.5.2. Sección no cableada de largo alcance

Los servicios de los ITS que requieren comunicación desde o hacia un móvil se pueden dividir en dos grandes ramas según el tamaño del área de cobertura asociada: comunicaciones de largo alcance y comunicaciones de corto alcance. Las primeras difunden la información dentro de su área a todos los móviles, a uno o a un grupo de ellos, y también comunican datos desde los móviles en dicha área hacia un centro de gestión.

Una de las soluciones para áreas urbanas y movilidad reducida son las MAN no cableadas para servicio de usuarios sin conexión cableada, organizadas en microcélulas de diámetro inferior a un kilómetro. El sistema es menos complejo que los sistemas celulares, pero también menos flexible, utiliza conmutación de paquetes, siendo su posible uso en ITS la comunicación con puntos fijos (domicilios particulares, oficinas, puestos de información) o con usuarios de poca movilidad (p.e. peatones) sin necesidades de tiempo real crítico.

Los sistemas celulares, por su área de cobertura adecuada a la movilidad de los vehículos, tienen un potencial de uso en los ITS muy destacado. En ellos, un equipo móvil terminal se comunica con la estación base correspondiente a la célula donde se encuentra el móvil. Las estaciones base de un área de servicio se comunican, por enlace cableado o por microondas, con un sistema intermedio que actúa de central de conmutación, pudiendo conectarse a otras estaciones intermedias del sistema celular o con redes centrales de conmutación de redes de usuarios fijos.

El más extendido de los sistemas celulares es el sistema GSM (*Global System for Mobile Communications*) un estándar paneuropeo (adoptado ya en 128 estados) para la transmisión digital de voz y datos operando en la región de 900 MHz. El sistema proporciona ocho canales por portadora con un espaciado entre portadoras de 200 KHz. Empleando un ancho de banda de 25 MHz, supone la existencia de 125 portadoras, esto es, 1000 canales de comunicación. Junto a los canales de voz y de servicios de conmutación, el sistema GSM permite la comunicación de paquetes de datos SMS (*Short Message Services*), paquetes limitados a 160 bytes, con comunicación en ambos sentidos entre el móvil y un centro de servicio.

Igualmente existen sistemas privados de comunicación por radio, principalmente sistemas para voz (sistemas de *trunking*) existiendo algunos sistemas de datos. Estos sistemas son de naturaleza propietaria, con importantes limitaciones en cuanto al área geográfica y al número de usuarios simultáneos, por lo cual son de uso limitado en el ámbito global de los ITS siendo sus posibles usos soluciones particulares para gestión de flotas o operación de vehículos comerciales de compañías privadas.

Los satélites para la comunicación de voz y datos, la mayoría de los cuales ofrece servicios de posicionamiento (el más importante de los cuales es el GPS, *Global Positioning System*). Los sistemas de comunicación por satélite proporcionan grandes áreas de cobertura, flexibilidad en la configuración de la red y capacidades especiales de difusión.

Sin embargo, los sistemas celulares presentan importantes ventajas respecto a los basados en satélite, como son:

- equipos terminales más pequeños y económicos, y tarifas más bajas;
- mejor cobertura en áreas urbanas o de edificación densa;
- menor tiempo de propagación y
- redes de mayor capacidad.

Los sistemas de difusión por radio (servicio unidireccional), que sustentan los sistemas de información en ITS, utilizan diversas técnicas, desde la más sencilla HAR (*Highway Advisory Radio*) hasta otras más elaboradas como el uso de subportadoras en RDS-TMC o el uso del procesamiento digital de señales en los DAB.

El sistema ALERT C (*Advice and problem Location for European Road Traffic*) de codificación de mensajes, uno de los resultados más importantes del programa europeo DRIVE, define el estándar RDS-TMC (*Radio Data System - Traffic Message Channel*) con capacidad de comunicación de 300 bps para la transmisión de mensajes relacionados con los incidentes y condiciones del tráfico.

La tecnología DAB (*Digital Audio Broadcasting*, o también *Data and Audio Broadcasting*) utiliza técnicas de proceso digital de señales para codificación y modulación y para compresión de audio, obteniendo calidad de CD en la recepción de audio, protección contra interferencias y uso optimizado del ancho de banda.

2.3.5.3. Sección no cableada de corto alcance

Ciertos servicios de los ITS precisan comunicaciones dedicadas de corto alcance (DSRC: *Dedicated Short Range Communication*) entre vehículos en movimiento y estaciones fijas (balizas): peaje automático, pago automático de aparcamientos, inspecciones de seguridad e identificación sin parada, etc. Dichas comunicaciones están intrínsecamente ligadas a áreas geográficas reducidas (hasta un radio máximo de unos 100 metros) con un número de usuarios simultáneos reducido y sin interferencias entre estaciones fijas

Existen tres tecnologías candidatas para las comunicaciones dedicadas de corto alcance: radiofrecuencia activa, radiofrecuencia pasiva e infrarrojos. Existen diversas implementaciones comerciales de comunicaciones DSRC que utilizan radiofrecuencia

activa, por lo general propietarias y no interoperables. El principal motor del desarrollo e implementación de las DSRC son los sistemas de peaje automático (ETTM). Una de las propuestas de estandarización en Europa utiliza radiofrecuencia activa por microondas a 5.8 GHz.

Los sistemas por radiofrecuencia pasiva utilizan diferentes frecuencias entre 902 y 928 MHz para evitar interferencias entre los receptores. Por su parte los sistemas de balizas con comunicación por infrarrojos, cuya propuesta de estandarización apunta a una longitud de onda de 850 nm, están limitados en su capacidad de soporte de grandes flujos de datos.

Los sistemas de radiofrecuencia activa son técnicamente superiores en prestaciones a los pasivos, que por su parte requieren un equipamiento de los vehículos menos costoso. Los sistemas de radiofrecuencia pasivos podrán ser utilizados en aplicaciones donde los vehículos circulan por un carril determinado con una velocidad limitada (instalando un detector por carril). En otras aplicaciones, como inspección en movimiento o comunicaciones (p.e. identificación y/o pago) con vehículos a gran velocidad, se precisan las mayores prestaciones técnicas de los sistemas activos.

Las futuribles autopistas automatizadas (AHS: *Automated Highway Systems*) requieren además comunicaciones no cableadas de corto alcance entre vehículos en movimiento, comunicaciones que deberían ser compatibles con las comunicaciones entre móviles y estaciones fijas. Estas comunicaciones aún se encuentran en una fase de investigación primaria y, aunque están consideradas dentro de la arquitectura general, están muy lejos de la implementación generalizada. La investigación en este campo apunta al uso de altas frecuencias (63 GHz) para evitar el fenómeno de absorción del vapor de agua.

2.4. ESTÁNDARES EN TELEMÁTICA DEL TRANSPORTE

Los sistemas de transporte inteligentes (ITS) introducirán multitud de nuevos sistemas y dispositivos en la actual infraestructura de transporte y control de tráfico. A medida que los sistemas aumentan en complejidad y sofisticación, planificadores, usuarios y fabricantes de equipamiento deberán reconocer la necesidad de permitir la interoperatividad entre sistemas mediante la aceptación de unos estándares.

2.4.1. IMPACTO SOCIOECONÓMICO DE LA ESTANDARIZACIÓN

Anteriormente a los intentos de estandarización (lo cual quiere decir que es válido para la gran mayoría de las instalaciones existentes) cada fabricante de equipos empleados en gestión de tráfico y transporte empleaba un protocolo de comunicación de datos diferente. Esto hacía muy difícil y económicamente prohibitivo integrar equipamiento de diferentes procedencias en un mismo entorno, entorpeciendo por otro lado la interacción entre ciudades o áreas adyacentes. La consecuencia de esto es que el explotador (la autoridad pública competente en la materia) se veía virtualmente obligado a contratar toda su infraestructura a un mismo fabricante o empresa, siendo extremadamente complejo realizar cualquier modificación o adición al margen de dicha compañía. Un conjunto de estándares común, si es seguido por los fabricantes y suministradores, puede romper esta tendencia en beneficio de los usuarios.

2.4.2. ORGANISMOS INTERNACIONALES DE ESTANDARIZACIÓN

El análisis del trabajo del Comité Europeo para la Normalización CEN en su Comité Técnico TC 278 "Telemática del Tráfico y Transporte por Carretera", nos permitirá sacar algunas conclusiones. Este comité abarca los 13 grupos siguientes

CEN - TC 278 - Telemática del Tráfico y Transporte por Carretera

WG1. Peaje automático y Control de acceso

WG2. Sistemas de Gestión de Mercancías y Flotas

WG3. Transporte Público

WG4. Información sobre tráfico y transporte

WG5. Control de Tráfico

WG6. Gestión de aparcamiento

WG7. Bases de datos geográficas de rutas

WG8. Datos de tráfico rodado: elaboración, almacenamiento y distribución

WG9. Comunicaciones dedicadas de corto alcance

WG10. Interfaces hombre-máquina

WG11. Interfaces subsistema y entre sistemas

WG12. Identificación automática de vehículos y equipos

WG13. Arquitectura y Terminología

Por su parte, el Comité Técnico TC 204 "Sistemas de Control e Información en Transporte" de la Organización Internacional para la Estandarización ISO, equivalente en dicho organismo al anteriormente referido CEN/TC 278, tiene como grupos de trabajo los siguientes:

ISO TC 204 - Sistemas de Control e Información en Transporte

WG1. Arquitectura y Terminología.

WG2. Requisitos de fiabilidad y calidad

WG3. Bases de datos de Tráfico y Transporte

WG5. Sistemas de peaje

WG6. Gestión de flotas

WG7. Gestión comercial de mercancías

WG8. Transporte Público y Emergencias

WG9. Información, gestión y control de tráfico integrados

WG10. Sistemas de información al viajero.

WG11. Sistemas de navegación y guiado de rutas

WG12. Gestión de aparcamientos y *off-road*

WG13. Interfaz hombre-máquina y factores humanos

WG14. Sistemas de control y advertencia vehículo/vía

WG15. Comunicaciones dedicadas de corto alcance

WG16. Comunicaciones de largo alcance / protocolos e interfaces.

El análisis de la relación de grupos de trabajo y de los contenidos de cada uno de ellos permite confirmar el hecho de que en dichos organismos (y por tanto en gran parte del mundo donde se emplean los ITS) apenas sí se ha prestado atención a los protocolos de los niveles inferiores excepto en nichos específicos como las DSRC.

2.4.3. VISIÓN ESTADOUNIDENSE DE LA ELABORACIÓN DE ESTÁNDARES DE COMUNICACIÓN

Los estándares, actualmente en desarrollo en diversas instancias de diversos países, consisten, más que en único protocolo, en una familia de protocolos que proporcionen interoperatividad entre servicios de control de tráfico y adquisición de datos con la posibilidad de hacerlo para distintas topologías y distintas necesidades de encaminamiento de datos. Esta familia de protocolos trata de cubrir varias alternativas de sistemas de comunicación y capacidades de los mismos, de una manera que permita modificar las capacidades de un sistema con alteraciones mínimas [NTCIP96f].

La elaboración de los estándares sigue el modelo de referencia OSI (Open System Interconnect Reference Model) que define siete clases diferentes de procedimientos para asegurar el intercambio de datos; estas clases son las conocidas capas o niveles. La combinación de dichas capas es referida en ocasiones como pila de protocolos. El modelo trata de ser exhaustivo, dado que no todos los sistemas precisan de las siete capas, y en ocasiones los procedimientos de algunas capas se funden en una sola.

2.4.3.1. Perfiles

En la descripción de estándares para comunicación en tráfico, se llama "perfil" a un conjunto específico de definiciones para las distintas capas necesario para describir la pila de protocolos. Dado que ningún protocolo de comunicación puede ser válido para todos los posibles condiciones y requisitos de comunicación, la descripción de un protocolo estándar va acompañada de un conjunto de variaciones u opciones aprobadas, definiéndose los diferentes perfiles.

2.4.3.2. NTCIP

Hasta fechas muy próximas, no existía ningún protocolo estándar para indicar como debían comunicarse unos componentes con otros dentro de las instalaciones de ITS , resultando que cada fabricante desarrollaba su propio protocolo para satisfacer sus necesidades. Para integrar sistemas fabricados por diferentes fabricantes los costes de desarrollar software específico para permitir esta interrelación son muy altos y, en ocasiones, es la integración es imposible y deben mantenerse infraestructuras físicas diferentes para dispositivos de distintas procedencias. Ante este problema creciente con el desarrollo de los ITS ante el cual la FHWA estadounidense patrocinó en 1993 la primera reunión de fabricantes de equipamiento de señalización de tráfico en la cual, la proliferación de protocolos de comunicación incompatibles fue identificada como una de

las principales barreras en el avance de las ITS. En consecuencia, la FHWA emprendió como tarea prioritaria el desarrollo de un protocolo de comunicación no propietario para ITS.

De hecho, desde los años 70, la NEMA estadounidense (*National Electrical Manufacturers Association*) introdujo un primer estándar no completo de conexiones para controladores de señalización de tráfico. En la década de los 80 algunas legislaciones locales propusieron protocolos de obligado seguimiento y en la década actual esta necesidad de una adecuada estandarización de las comunicaciones no ha hecho más que incrementarse.

NEMA, en consecuencia, ha desarrollado el NTCIP (*National Traffic Control IVHS communication Protocol*). Este protocolo pretende emplear las mejores características de los estándares existentes en el mundo de las comunicaciones, basándose en el modelo OSI de siete niveles adoptado como modelo de referencia por la ISO.

Cualquier estándar global de comunicaciones que se establezca, debería cumplir los siguientes requisitos:

a) posibilidad de sustitución de dispositivos por otros del mismo tipo pero diferente procedencia (fabricante);

b) conectividad entre dispositivos del mismo tipo y diferente procedencia;

c) interoperatividad entre distintos dispositivos de control de tráfico comunicándose con un mismo centro de control de tráfico, empleando una única infraestructura de comunicaciones de uso común a todos ellos;

d) comunicación entre centros de control;

e) posibilidad de integrar futuras tecnologías con un impacto mínimo (preferiblemente nulo) sobre los sistemas existentes.

Un típico ejemplo de entorno de aplicación para un estándar de este tipo es un computador central en una sala de control de tráfico local que monitoriza y dirige el funcionamiento de reguladores (controladores basados en microprocesadores) que gestionan los semáforos de una ciudad. El computador central envía instrucciones periódicas a los reguladores para modificar los ciclos y repartos según las condiciones del tráfico. Los reguladores por su parte, pueden recibir datos de sistemas de detección o simple captación de datos y envían al computador central informaciones sobre flujos de tráfico u otras informaciones.

2.4.4. ESTADO ACTUAL DE DEFINICIÓN DE ESTÁNDARES NTCIP

Para las aplicaciones y entornos de gestión de tráfico y transporte, en sus requisitos de comunicaciones, se han definido estándares en cinco de las capas del modelo de referencia OSI, que pueden observarse en la tabla de la figura 2.12.

- capa física: comúnmente se emplea RS-232 y modems FSK.
- capa de enlace de datos: se emplean el protocolo punto a punto (PPP) y el protocolo punto a multipunto (PMPP); ambos se basan en HDLC.
- capa de red: cuando se utilice, se empleará IP.
- capa de transporte: cuando se utilice, se empleará UDP o TCP.
- capa de sesión y presentación: no se emplean.
- capa de aplicación: puede emplearse TELNET, FTP, STMP (protocolo de gestión de transporte) o SNMP (protocolo de gestión de red).

	PERFILES DEL NTCIP			
niveles	CLASE B	CLASE A	CLASE C	CLASE E
APLICACIÓN	STMF	STMF	SNMP FTP / TELNET	SNMP FTP / TELNET
PRESENTACIÓN	nulo	nulo	nulo	nulo
SESIÓN	nulo	nulo	nulo	nulo
TRANSPORTE	nulo	UDP	TCP	TCP
RED	nulo	IP	IP	IP
ENLACE DE DATOS	PMPP	PMPP	PMPP	PPP
FISICO	EIA232E FSK	EIA232E FSK	EIA232E FSK	EIA232E

Fig. 2.12. *Perfiles definidos por el NTCIP*

2.4.4.1. Perfiles de comunicación en tráfico

Actualmente se definen cuatro perfiles, a los que se puede añadir alguno más. Los cuatro perfiles actualmente definidos son:

Perfil de Clase A: intercambio de información entre dispositivos en el mismo enlace o en enlaces diferentes.

Perfil de Clase B: intercambio de información en tiempo real crítico entre una estación primaria y estaciones secundarias en un mismo enlace.

Perfil de Clase C: intercambio de información entre dispositivos en el mismo enlace o en enlaces diferentes, con secuenciamiento y transferencia fiable que permita transferencia de ficheros eficiente.

Perfil de Clase E: intercambio de información entre dispositivos en el mismo enlace o en enlaces diferentes asumiendo una naturaleza punto a punto, en lugar de la línea de difusión. Este perfil también incluye secuenciamiento y transferencia fiable.

2.4.4.2. Ámbitos de aplicación de los perfiles

La clase A permite el intercambio de datos entre dispositivos y controladores conectados mediante un controlador intermedio. Es un servicio sin conexión, en el sentido de que no es necesaria ninguna preparación de la línea previa a la transmisión. Emplea protocolos de transporte "no fiable" (el nivel de transporte emplea UDP), quedando la fiabilidad de la entrega fuera del ámbito del protocolo, debiendo responsabilizarse la aplicación de la detección de errores y de la recuperación, en su caso. De este modo, en esta clase se da preferencia a la velocidad de la transmisión sobre la fiabilidad de la misma. Las capas bajas utilizan el protocolo punto-multipunto, con múltiples dispositivos residiendo en una misma línea de comunicación (red de difusión). Se emplea en transferencia de datos que pueda necesitar encaminamiento; no obstante, esta capacidad introduce un sobrecoste importante.

La pila de protocolos de clase B es la más sencilla posible: tan sólo contempla protocolos en el nivel físico, nivel de enlace de datos y nivel de aplicación; como en la clase A, no se preocupa por la fiabilidad de la entrega. La clase B tampoco contempla el encaminamiento (no tiene capa de red), luego se límita a dispositivos conectados directamente, sin estaciones intermedias. Esta clase de Perfil está dedicada a hacer *"polling"*, con mensajes de orden y respuesta dando prioridad al tiempo de envío. La principal ventaja de la clase B es la reducción de la información adicional (*overhead*) contenida en cada mensaje, mejorando la velocidad de transmisión.

El perfil de clase C proporciona servicios orientados a conexión, significando que toda transmisión requiere un periodo de preparación para establecer la comunicación. Esto reduce la eficiencia global, pero permite la fiabilidad en las transmisiones (la capa de

transporte es en este caso TCP -*Transmission Control Protocol*- y garantiza la fiabilidad de los datos entregados a la aplicación, liberando a ésta de la detección de errores en la mayoría de casos). Por el sobrecoste introducido en cada mensaje, no resulta el más apropiado para líneas de baja velocidad. Sus aplicaciones más comunes son entre dispositivos que requieran intercambio fiable y en ocasiones de grandes ficheros, como servicios de información por radio o, en algunos casos, paneles de mensajes variables. En el nivel de aplicación emplean tanto SNMP, como aplicaciones como Telnet o FTP.

La clase E proporciona servicios equivalentes a los de clase C, pero sobre una topología de naturaleza punto a punto, en lugar de la punto-multipunto empleado en los otros tres perfiles anteriores. El nivel físico no es de difusión, y el nivel de enlace de datos es el protocolo Punto a Punto. Esta clase tiene una orientación a la comunicación entre centros de tráfico, aunque nada impide su utilización en otros entornos. Tanto la clase C como la clase E incluyen, dentro de sus servicios de comunicación fiable, procesos de autentificación que permitan identificar con seguridad el origen y el destinatario de un mensaje.

2.4.4.3. Futuros perfiles

Los perfiles anteriormente expuestos no contemplan todavía algunas de las últimas tendencias. Por ejemplo, los interfaces de fibra óptica no están incluidos, lo cual no debe impedir integrarlos en sistemas de comunicación siempre que sigan los protocolos anteriores como un mínimo para garantizar la compatibilidad, permite mayor velocidad en determinadas partes de la red.

Por otra parte, dentro de los perfiles citados y en la mayoría de las aplicaciones actuales en sistemas de tráfico, la comunicación tiende a organizarse con una estación primaria y una serie de estaciones secundarias. El uso de comunicaciones balanceadas (*peer-to-peer*) en medios más rápidos y con mecanismos de control distribuidos es una tendencia en alza, con el uso de protocolos como Token Ring o Aloha, pero la selección de unos de ellos para definir un perfil aun está en estudio, dependiendo del medio físico empleado, referido a los últimos avances (fibra óptica, cable coaxial, radio) y al tipo de mensajes empleados; parece razonable pensar que, por su diversidad, cada medio físico pueda definir un perfil diferente.

Otro tipo de perfil que pudiera llegar a incorporarse son las conexiones bajo demanda (*dial-up access*) cuando las comunicaciones a tiempo completo suponen un coste innecesario. Un ejemplo de entornos de aplicación serían dispositivos remotos contadores de tráfico o medidores ambientales, que requieren normalmente comunicaciones periódicas o esporádicas sin restricciones de tiempo real. Son elementos importantes en este caso la

transferencia de ficheros y la autentificación de los extremos. Un perfil para este tipo de comunicaciones permitiría estandarizar este tipo de intercambio.

2.4.4.4. Perfil de clase B

Como se ha indicado, el perfil de clase B pretende cubrir la necesidad de un protocolo específico para los requisitos de comunicación de dispositivos de campo como reguladores de tráfico, señalización variable, sistemas de videocámara y dispositivos similares conectados directamente a un dispositivo central (no hay encaminamiento). Este perfil es, por diseño, implementable con dispositivos actualmente en uso, y está orientado a intercambio de información con restricciones de tiempo real entre dispositivos de campo y controladores conectados directamente a través de un enlace de comunicación [NTCIP96b].

Los protocolos definidos en el perfil de clase B facilitan la conexión y el control de dispositivos de campo incluso aquellos con baja capacidad de procesamiento o que sólo soporten bajas velocidades de comunicación. No están ligados a ninguna arquitectura específica; únicamente asumen una organización estación primaria / estación secundaria con conexión directa entre ambas (y ninguna capacidad de encaminamiento). Los protocolos escogidos se caracterizan por su adaptabilidad a enlaces de baja velocidad (sin perjuicio para las altas velocidades).

Nivel físico: puede emplear EIA/TIA-232-E o módem FSK. En el primer caso, el dispositivo emplea un interface EIA/TIA-232-E con un mínimo de 1200 bps, transmisión asíncrona, 1 bit de arranque, 8 de datos, sin paridad y 1 bit de parada. Debe disponer de un conector hembra de 25 pines. En el caso del módem FSK, puede emplear half-duplex de 2 hilos o full-duplex de 4 hilos sobre un canal privado, técnicas de multiplexación por división de tiempos, transmisión por modulación FSK (*phase coherent Frequency Shift Keying*) a 1200 bps. El formato de datos debe ser asíncrono y en serie. El conector será macho de 9 pines.

Nivel de enlace de datos: el nivel de enlace de datos emplea el protocolo punto-multipunto descrito en [NTCIP-PMPP] . El identificador de protocolo superior será el correspondiente a STMF (0xC1).

Nivel de red, nivel de transporte, niveles de sesión y presentación: nulos.

Nivel de aplicación: será conforme al STMF (*Simple Transportation Management Protocol*) descrito en NEMA TS-3.2 como una variación de SNMP (*Simple Network Management Protocol*) descrito primeramente en [RFC-1157].

2.4.4.5. Perfil de clase A

El perfil de clase A se dirige a cubrir la necesidad de un protocolo específico para los requisitos de comunicaciones de dispositivos de campo como reguladores, señalización variable, controladores de cámaras, ..., que pueden no estar directamente conectados a otro elemento que precisa comunicarse con ellos (en consecuencia necesitan la posibilidad de encaminamiento de datos) [NTCIP96a].

Por ello, la pila de protocolos de la clase A es apropiada para el intercambio de información entre dispositivos de campo y controladores que pueden estar en diferentes subredes; la característica de la clase A respecto a la clase B es el soporte del encaminamiento. La clase A permite su uso en dispositivos con suficiente capacidad de procesamiento y capacidad de comunicación como para soportar un protocolo con un cierto sobrecoste (*overhead*) debido a dicha capacidad de encaminamiento. Para esta clase también se asume una relación estación primaria / estación secundaria. Este perfil puede funcionar en enlaces de comunicación de baja velocidad, aunque el *overhead* necesario limita en la práctica su uso en enlaces excesivamente lentos.

En resumen, la clase A se emplea en aplicaciones donde se precise intercambio sin restricciones de tiempo real y la capacidad de encaminamiento entre subredes.

Nivel físico: el mismo que en la clase B.

Nivel de enlace de datos: el nivel de enlace de datos emplea el protocolo punto-multipunto descrito en [NTCIP-PMPP]. El identificador de protocolo superior será el correspondiente a IP (0x21).

Nivel de red: será conforme al protocolo Internet IP, sin opciones, y con UDP (0x11) en el nivel superior.

Nivel de transporte: se implementará el protocolo de datagramas de usuario (UDP).

Niveles de sesión y presentación: nulos.

Nivel de aplicación: el mismo que en la clase B.

2.4.4.6. Perfil de clase C

El perfil de clase C se dirige a cubrir la necesidad de un protocolo específico para los requisitos de comunicaciones de dispositivos de campo "avanzados" que puede emplear transferencia de ficheros [NTCIP96c].

Por ello, la pila de protocolos de la clase A es apropiada para el intercambio fiable de información entre dispositivos de campo y controladores que pueden estar en diferentes subredes; la característica de la clase C es el soporte del encaminamiento y la transferencia fiable de datos. La clase C permite su uso en dispositivos con suficiente capacidad de procesamiento y capacidad de comunicación como para soportar un protocolo con un cierto sobrecoste debido a la capacidad de encaminamiento y la exigencia de fiabilidad en la transferencia. Para esta clase también se asume una relación estación primaria / estación secundaria. Este perfil puede funcionar en enlaces de comunicación de baja velocidad, aunque el sobrecoste limita en la práctica su uso en enlaces excesivamente lentos.

Nivel físico: el mismo que en la clase B.

Nivel de enlace de datos: el nivel de enlace de datos emplea el protocolo punto-multipunto descrito en [NTCIP-PMPP] . El identificador de protocolo superior será el correspondiente a IP (0x21).

Nivel de red: será conforme al protocolo Internet IP, sin opciones, y con TCP (0x06) en el nivel superior.

Nivel de transporte: se implementará el protocolo de control de la transmisión (TCP).

Niveles de sesión y presentación: nulos.

Nivel de aplicación: será conforme al STMF y también conforme a FTP (*File Transfer Protocol*).

2.4.4.7. Perfil de clase E

El perfil de clase E se dirige a cubrir la necesidad de un protocolo específico para los requisitos de comunicaciones entre centros de gestión y control de tráfico y/o emergencias.

Por ello, la pila de protocolos de la clase E es apropiada para el intercambio fiable de información entre dispositivos de campo y controladores que puedan estar conectados a redes públicas; la característica de la clase E es el soporte del encaminamiento y la transferencia fiable de datos con seguridad en la comunicación [NTCIP96e]. La clase E permite su uso en dispositivos con suficiente capacidad de procesamiento y capacidad de comunicación como para soportar un protocolo con un cierto sobrecoste sobre enlaces punto a punto. Este perfil puede funcionar en enlaces de comunicación de baja velocidad, aunque el sobrecoste limita en la práctica su uso en enlaces excesivamente lentos.

Nivel físico: el mismo que en la clase B.

Nivel de enlace de datos: el nivel de enlace de datos emplea el protocolo punto-a-punto descrito en [RFC1661 y RFC1662]. El identificador de protocolo superior será el correspondiente a IP (0x21).

Nivel de red: será conforme al protocolo Internet IP, sin opciones, y con TCP (0x06) en el nivel superior.

Nivel de transporte: se implementará el protocolo de control de la transmisión (TCP).

Niveles de sesión y presentación: nulos.

Nivel de aplicación: los mismos que en la clase C.

2.5. CONCLUSIONES

Como se puede deducir de lo expuesto en este capítulo, los desarrolladores e implementadores de ITS han dedicado escasa atención a los niveles inferiores de la comunicación, excepto en el caso de los medios físicos específicos de las ITS como son las Comunicaciones Dedicadas de Corto Alcance (DSRC) o los distintos medios de comunicaciones no cableadas de largo alcance.

La visión extendida de que los niveles inferiores del modelo de referencia OSI no atañen a los desarrolladores de ITS ha conducido a dos visiones muy diferentes: en algunas ocasiones se desarrollan sistemas debido a intereses comerciales en los cuales se emplean protocolos 'propietarios' cerrados, mientras que en otros casos se limitan a emplear protocolos existentes sin estudiar su adecuación a las necesidades de los ITS y sin llegar a determinar un estándar para estos sistemas.

Se ha revisado en profundidad el importante trabajo llevado a cabo en la definición de protocolos para la capa de aplicación, y sólo en los casos específicos citados se ha trabajado en protocolos de los niveles bajos (los que hemos llamado nivel de la comunicación en la arquitectura de las comunicaciones para ITS). Como confirmación a una de las hipótesis iniciales que dieron lugar a esta memoria, la excepción más reseñable a esta tendencia ha sido la especificación del protocolo punto-multipunto y el resto de especificaciones seleccionadas dentro del NTCIP.

La segunda consecuencia de la revisión corrobora que, en caso de existir aplicaciones de los ITS con especiales requisitos de fiabilidad, estos aspectos de fiabilidad de los

dispositivos propios de los ITS y de la fiabilidad de la comunicación desde y hacia ellos no han sido tratados explícitamente.

La revisión permite afirmar que no existe ninguna propuesta específica de comunicaciones fiables. En caso de querer establecer comunicaciones con un mayor grado de fiabilidad, éstas deben implementarse mediante una red paralela específica para aplicaciones de fiabilidad crítica, ya que no existe la previsión de que los enlaces fiables puedan integrarse en la estructura general de las redes de comunicaciones para los ITS, coexistiendo enlaces fiables con enlaces que no lo sean.

CAPÍTULO TERCERO

REQUISITOS DE FIABILIDAD DE LAS COMUNICACIONES EN ITS

El esfuerzo en la aplicación de las tecnologías avanzadas de la información y de las comunicaciones en el sector del transporte y del tráfico se ha centrado en ofrecer oportunidades en un buen número de temas críticos. Sin embargo, la utilización de estas tecnologías plantea nuevos problemas a los cuales hay que dar respuesta: asegurar la necesaria interoperatividad de los equipos, la comprensión por parte de los usuarios de los servicios puestos a su disposición, el desarrollo de interfaces entre los distintos modos de transporte y conseguir su utilización en todas las regiones para conseguir una continuidad espacial de los servicios.

El esfuerzo en todas estas líneas ha postergado un análisis más detallado de la fiabilidad de los sistemas de ITS y los mecanismos de tolerancia a fallos aplicables en los mismos, particularmente en las comunicaciones empleadas en los ITS. Este capítulo tratará de confirmar y delimitar las posibles carencias existentes en las actuales propuestas de estandarización para las comunicaciones en los ITS respecto a la fiabilidad de las mismas.

Por ello es necesario determinar, como se hará en esta memoria, qué aplicaciones de los ITS van a precisar comunicaciones especialmente fiables, cuáles son sus requisitos de fiabilidad, cuáles son los procedimientos de tolerancia a fallos que permiten alcanzar estos requisitos de fiabilidad y como deben integrarse estas propuestas de algoritmos y procedimientos para la tolerancia a fallos que aumenten la fiabilidad de la comunicación dentro de las propuestas de estandarización para las comunicaciones en ITS sin necesidad de modificar la arquitectura propuesta, con recursos extraídos de la tecnología de comunicaciones de uso común.

Por su nivel de desarrollo e implantación, por sus especiales requisitos de fiabilidad y por el conocimiento corporativo que de ellos se tiene, de entre las áreas funcionales de los ITS el trabajo se enfoca a los Sistemas de Gestión y Control del Tráfico, que se basan en comunicaciones entre estaciones fijas con redes cableadas, y dentro de dichas redes

cableadas se enfocará la investigación en la red local (ver figura 2.10) específicamente instalada para los ITS, con especial atención a los enlaces punto a punto por considerar que ofrecen una mayor versatilidad y permiten diseñar sistemas con propiedades de fiabilidad suficientes, dentro de las limitaciones funcionales, ambientales y económicas de los sistemas de comunicación aplicables.

En cualquier caso, una de las preocupaciones del explotador de una sistema ITS es el grado de escepticismo con que el público recibe un nuevo servicio y cómo la aceptación de éste depende de la credibilidad del mismo. Esto supone que el sistema debe estar funcionando adecuadamente el 99,99 % del tiempo, lo cual implica unas características de fiabilidad y disponibilidad ante la posible aparición de fallos inevitables en un sistema de campo. Como declaraba un Jefe de Tráfico de una gran ciudad americana : "No podemos estar llamando a alguien a medianoche y decirle que vaya a la calle a arreglarlo inmediatamente". Es necesario proveer al sistema de la tolerancia a fallos apropiada. "Si un sistema ITS pierde la credibilidad, no creo que pueda recuperarla" afirmaba el mismo experto [Wer96].

Debe considerarse así mismo que una vez en funcionamiento, si el servicio ha recibido un grado de aceptación adecuado, el público espera el mantenimiento y mejora del mismo sin interrupciones. Un plan de mantenimiento adecuado e integrado con la provisión de tolerancia a fallos debe contemplar el cumplimiento de ese objetivo.

Este capítulo tratará en primer lugar los requisitos de fiabilidad de los sistemas de control de tráfico para en segundo término analizar la fiabilidad en las comunicaciones de los sistemas ITS existentes y la que proporciona el uso de los estándares propuestos para la comunicación de los ITS.

3.1. FIABILIDAD DE LOS SISTEMAS DE CONTROL DE TRÁFICO

Los sistemas de gestión y control del tráfico constituyen un área funcional de especial relevancia dentro de los ITS, tanto por la importancia como por el grado de criticidad de sus efectos. En el análisis de aplicaciones relacionadas con control de tráfico, tanto urbano como interurbano, los requisitos de comunicaciones deben considerarse no sólo en cantidad de información a transmitir y tiempo de respuesta, sino también según la necesidad de la recepción de los datos en tiempo debido y los efectos de pérdida o corrupción de los datos.

Para estudiar estos requisitos habrá que analizar tanto las aplicaciones de control de tráfico en particular con los elementos de los ITS implicados en ellas como las redes de comunicaciones sobre las que se implementan, concentrando primeramente el estudio en las redes locales (donde las comunicaciones son directas y por tanto sin posibilidades de encaminamiento) para establecer después algunas consideraciones sobre las redes de nivel superior.

3.1.1. ESTRUCTURA DE UN SISTEMA DE CONTROL DE TRÁFICO

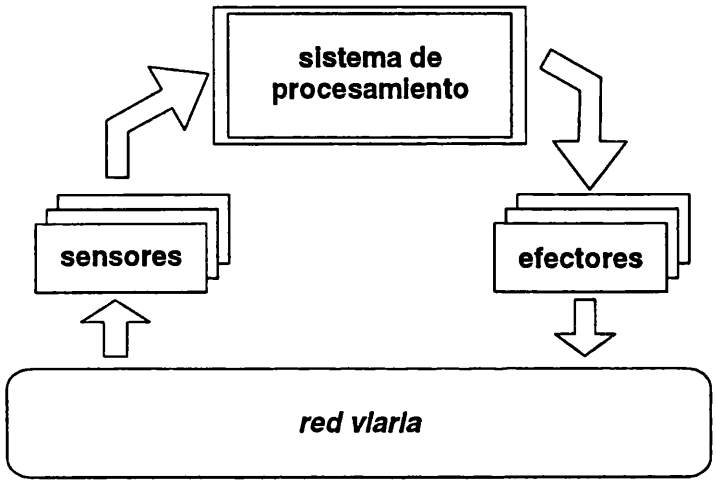


Fig. 3.1 *Estructura de un sistema de control de tráfico*

Un sistema de control de tráfico constituye un tipo de sistemas de control donde la realidad física que se trata son las condiciones de la red viaria. El sistema consiste en un lazo cerrado, representado en la figura 3.1, formado por

- la red viaria;
- un conjunto de dispositivos de sensorización, situados en la red viaria, que toman medidas de las condiciones o estado de la misma;
- un conjunto de efectores o medios de intervención sobre la red viaria (por lo general bastante limitados);

- un sistema de procesamiento cuya función es seleccionar convenientemente las posibles acciones usando la información disponible proveniente de los sensores para tratar de corregir el comportamiento del sistema;
- un sistema de comunicación entre los sensores, el sistema de procesamiento y los efectores.

Los distintos elementos, especialmente los situados a la intemperie (que son la gran mayoría) son susceptibles de fallar, por lo cual deben existir mecanismos para sustentar la fiabilidad de los mismos. Los mecanismos de tolerancia a fallos que deben sustentar esta fiabilidad son de dos tipos:

- tolerancia a fallos de los dispositivos
- tolerancia a fallos de los enlaces de comunicaciones

Los apartados siguientes describen los elementos que constituyen los sistemas de control de tráfico, tratando la fiabilidad y tolerancia a fallos en los mismos y analizando los aspectos relacionados con las comunicaciones entre ellos.

3.1.2. SISTEMA DE PROCESAMIENTO

El procesamiento dentro de los sistemas de control de tráfico, consistente en el análisis, integración y combinación de las medidas recogidas durante la sensorización, con el fin de incrementar la calidad y la utilidad de la información para el control de tráfico, y a partir de allí tomar decisiones sobre los efectores, se realizaba en los primeros sistemas de control de tráfico por un único nodo central responsable de la gestión de toda la red viaria. Sin embargo, tal estructura constituye un punto crítico del sistema, de tal modo que un fallo en el centro de control o en las comunicaciones con el mismo produce una avería general del sistema.

La solución inmediata a dicho problema es la implementación de sistemas centrales de procesamiento más sofisticados, con redundancia de recursos y mecanismos de tolerancia a fallos en el computador central. Sin embargo, el celo excesivo en la fiabilidad del sistema central de procesamiento es vano en caso de que éste no pueda comunicarse con el exterior.

La tendencia más destacada en los sistemas de procesamiento para sistemas de control de tráfico son las estructuras piramidales, constituidas por un sistema de procesamiento

distribuido y jerarquizado, donde cada nodo de procesamiento controla una parte de la red bajo control de los niveles superiores, pero pudiendo funcionar de modo autónomo cuando fallan los niveles superiores o la comunicación con los mismos. La estructura piramidal del sistema de procesamiento en los ITS tiene su base en los reguladores que controlan los elementos dentro de una red local, según se muestra en la figura 3.2.

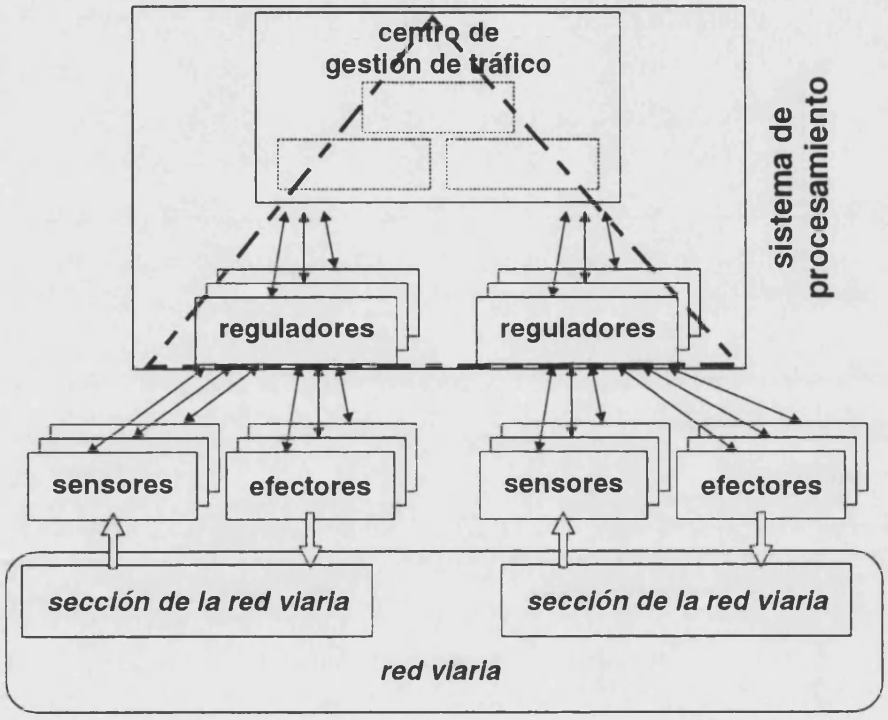


Fig. 3.2 *Estructura piramidal del sistema de procesamiento*

Así pues, las decisiones de actuación sobre la red viaria son adoptadas por el sistema central, que por medio de los reguladores dirige la actuación de los efectores que materializan dichas decisiones. En caso de que un regulador pierda la comunicación con el sistema central, es el propio regulador el que toma las decisiones y dirige la actuación de los efectores controlando un segmento de la red viaria.

3.1.3. SENSORES

Los sensores traducen una información (presencia, paso de vehículos, velocidad, dimensiones, etc.) en una señal eléctrica elemental representativa del parámetro que se

pretende medir. En la actualidad existen instalaciones con sensorización más o menos avanzada, cuyos datos pueden ser integrados y ser o no utilizados en una estrategia de control. Lo que tenemos es un buen recolector de datos, que nos permitirá: por un lado obtener una buena base estadística, y por otro el conocimiento del estado del tráfico, que puede ser utilizado para llevar a cabo una política de recomendación de itinerarios.

Existe una amplia gama de tecnologías de sensorización [Coh91], desde las más antiguas como los neumáticos de caucho (permiten contar el número de vehículos), los lazos electromagnéticos (cuentan el número de vehículos, calculan la tasa de ocupación y velocidad) y la fotografía aérea (permite obtener itinerarios, tiempos de recorrido, concentración, velocidad media, flujo, longitud de las filas de espera y giros), a las más recientes como el ultrasonido (presencia de vehículos), los radares (velocidad instantánea), infrarrojos (presencia) y el vehículo flotante (datos de flujo, concentración, velocidad media y tiempos de recorrido).

El sensor vídeo merece un interés especial [Mar95]. El principio básico de estos dispositivos es usar cámaras de vídeo, colocadas en la infraestructura, para después procesarlas en tiempo real mediante técnicas de tratamiento de imágenes (Visión Artificial). Las principales ventajas del uso de esta técnica son: posibilidad de instalación fija o móvil (permite incluso el procesado a posteriori) y acceso a nuevas medidas de tráfico. A parte de las medidas tradicionales, como son los volúmenes de tráfico, velocidades y tiempos de ocupación, permite obtener la distancia entre vehículos, longitudes de cola, densidad, porcentaje espacial ocupado, así como fenómenos relacionados con el comportamiento como son: la localización y cuantificación de los cambios de carriles, aceleraciones y deceleraciones en carreteras. Mención especial requiere la detección de matriculas para vigilancia y pago automático.

Los sistemas de control de tráfico están preparados para tolerar fallos del sistema de sensorización. En caso de fallo en alguno de los sensores o en la comunicación con los mismos, el sistema es capaz de trabajar con información incompleta tomando decisiones a partir de los datos existentes.

3.1.3.1. COMUNICACIÓN CON LOS SENSORES

Habitualmente los datos captados por los sensores no son transmitidos en el mismo instante que se captan, sino que son almacenados en memoria local y transmitidos al regulador posteriormente. Este almacenamiento permite procesar los datos, de modo que lo

que se transmite al regulador, en lugar de ser una mera acumulación de datos captados, suele ser una información elaborada a partir de los mismos (por tanto con un menor tamaño en bytes que disminuye el tiempo de transmisión necesario).

El protocolo empleado para la comunicación debe ser capaz de detectar si el bloque de bytes transmitidos sufre alguna alteración durante su transmisión. En tal caso, el bloque es corregido o descartado, según las posibilidades del protocolo empleado. Si la información es descartada, y se considera vital su recepción, el protocolo empleado debe ser capaz de reenviarla automáticamente. Igualmente, si el protocolo es capaz de detectar que se ha perdido un bloque de información deberá igualmente reenviarla si ello es posible.

Debe tenerse en cuenta que en las transmisiones provenientes de dispositivos de captación, algunas informaciones tienen una validez temporal, de tal manera que puede carecer de sentido retransmitir unos datos 'antiguos' para tomar decisiones. En tal caso, las decisiones se tomarían a falta de dicha información o se esperaría a que llegue la próxima información de dicho dispositivo.

Otro hecho a considerar en las comunicaciones con los dispositivos de captación es si el esquema de comunicación (el protocolo) da al dispositivo la posibilidad de iniciar una comunicación (protocolo equilibrado) o sólo responde a las peticiones provenientes de una estación primaria, el regulador (lo cual sería propio de una protocolo no balanceado). La diferencia entre el primer y segundo caso es que ante un situación de alarma generada por un dispositivo de detección de incidentes, sólo en una situación equilibrada es transmitida inmediatamente, mientras que en el otro caso depende de la frecuencia con la que la estación primaria interrogue a cada dispositivo (proceso de *polling*).

Excepto en los dispositivos de captación más primitivos, la comunicación es bidireccional para poder controlar el funcionamiento del dispositivo y modificar sus parámetros de funcionamiento. Esta bidireccionalidad permite asimismo utilizarla para protocolos que envíen reconocimientos positivos o negativos a los mensajes.

Los protocolos más empleados en la comunicación con los sistemas de captación son los no balanceados, aunque es conveniente emplear protocolos balanceados en la comunicación con los dispositivos más inteligentes de modo que estos sean capaces de iniciar la comunicación en caso necesario.

3.1.4. EFECTORES

El subsistema de efectores engloba el conjunto de recursos físicos que permiten la implementación de las estrategias de control. Son la llave para el sistema de control, ya que van a determinar el tipo de estrategia a considerar.

Como efectores básicos se encuentran los propios semáforos y la policía. En los últimos años se han desarrollado, para el entorno interurbano, los sistemas de paneles de mensaje variables (PMV), que permiten proporcionar al usuario una información, recomendación o incluso una orden. Las principales aplicaciones son: el control de la velocidad, advertencias de congestiones, guiado, regulación de adelantamientos, advertencias de condiciones meteorológicas adversas, tramos especiales (puentes, túneles), información sobre el estado de las carreteras adyacentes, recomendación de itinerarios alternativos, guiado a aparcamientos, información sobre atascos en horas punta, obras, carriles reversibles, ferias y acontecimientos deportivos, carriles especiales para uso por los transportes públicos, intersecciones y curvas peligrosas, pasos de montaña, cruces de ferrocarriles, cruce de peatones, entradas a poblaciones, áreas de escolares, así como en casos en los que podría haber riesgo de un inadecuado o peligroso comportamiento del conductor (entradas a curvas, o a áreas residenciales). El PMV en definitiva trata de regular el tráfico, informando al usuario y/o imponiendo restricciones u órdenes. Un ejemplo a gran escala de su uso (más de 250 PMV) lo constituye el proyecto SIRIUS para la regulación del tráfico en la región de París.

3.1.4.1. COMUNICACIÓN CON LOS EFECTORES

La comunicación con los efectores, siendo su uso primario el envío de órdenes hacia los dispositivos de señalización, debe ser obligatoriamente bidireccional para poder recibir el reconocimiento por parte del dispositivo de señalización de que éste ha recibido la orden correctamente. Igualmente la comunicación bidireccional puede utilizarse para interrogar al dispositivo para saber si éste está funcionando correctamente.

El conocimiento de que una orden de cambio en la señalización no ha sido recibida correctamente es en este caso fundamental, y en tal caso la orden debe ser reenviada. Además, el protocolo debe asegurar especialmente que los datos que forman la orden no han sido modificados durante la transmisión (por medio de secuencias de verificación de trama).

Por otra parte, si el dispositivo de señalización estuviera en una red abierta, por ejemplo un panel de mensajes variables comunicado a través de la red telefónica conmutada, se

hace necesario la utilización de un protocolo que autentifique la identidad de los nodos que se comunican para evitar intromisiones de nodos no autorizados. Esta necesidad de autentificación no está presente en las redes locales en ITS donde el regulador mismo es el que envía las órdenes al dispositivo de señalización y no encamina automáticamente paquetes hacia el dispositivo de señalización procedentes del exterior (las comunicaciones que deberían ser autentificadas son las recibidas por el regulador si éste se encontrase en una red abierta).

En caso de fallos en la comunicación, un dispositivo de señalización debe mantenerse en el mismo estado durante un cierto tiempo, pasado el cual sin recibir orden alguna pasa a un estado por defecto preestablecido.

Un panel de mensajes variables constituye un caso de dispositivo de señalización simple que ejemplifica los puntos expuestos al respecto.

Algunos dispositivos de señalización deben funcionar de forma coordinada, de modo que la señalización de cada uno de ellos debe ser adecuada y coherente en su conjunto. En este caso la regulación debe conocer la corrección o no de la comunicación a un dispositivo antes de emitir la comunicación a otro dispositivo coordinado con el primero. La regulación viene determinada por un autómata que indica la sucesión de comunicaciones que deben producirse y los procesos alternativos cuando la comunicación no es correcta.

Las consideraciones sobre los protocolos empleados en la comunicación son las ya indicadas para un dispositivo de señalización simple, siendo un autómata (que forma parte de la aplicación, y no de la red de comunicación) el que decide que comunicaciones deben producirse.

Cada uno de los dispositivos de señalización que forma un grupo coordinado tiene su propio modo de emergencia. Por su parte, el regulador que ejecuta el autómata recibe órdenes 'superiores' sobre los parámetros que rigen el conjunto (por ejemplo ciclos y repartos de un grupo semafórico). Si las comunicaciones con el regulador fallan, el modo de emergencia puede basarse en parámetros por defecto.

Un ejemplo de esta **señalización coordinada** lo constituye un conjunto de paneles de mensajes variables que vayan reduciendo la velocidad de manera progresiva según las condiciones del tráfico.

3.1.5. EFECTOS DE LOS FALLOS EN LAS COMUNICACIONES

Recogiendo lo expuesto anteriormente, las comunicaciones en las aplicaciones de control de tráfico se pueden considerar de manera distinta según se trate de comunicación con un sensor o con un dispositivo efector. En el caso de dispositivos de vigilancia y captación de datos pueden darse dos casos: dispositivos que transmiten periódicamente y dispositivos que transmiten sólo cuando detectan incidentes.

En el primer caso, si se produce un fallo transitorio su importancia dependerá del tiempo que transcurre entre una transmisión y otra (¿se puede esperar hasta recibir los datos de la situación de tráfico que llegarán en un tiempo T?) y de la necesidad o no de que llegue toda la información (¿el control de tráfico necesita la información captada en todos los momentos n*T?). En el caso de que una de las respuestas (o ambas) a las preguntas anteriores sea negativa, o en el caso de que la transmisión no sea periódica, la información que no llegue correctamente a su destino debería ser retransmitida.

En el caso de dispositivos de señalización y control donde la información transmitida son órdenes a los dispositivos, éstos deben confirmar la recepción correcta de las mismas (preferiblemente también la ejecución correcta) y dichas órdenes deben ser retransmitidas en caso de no llegar correctamente a su destino o no recibirse su confirmación (si es esperada).

Todo dispositivo partícipe en una aplicación ITS debe estar preparado para la eventualidad de un fallo en las comunicaciones que lo aíslen, temporal o definitivamente, del resto del sistema (al decir definitivamente quiere indicarse que el fallo persiste por tiempo indefinido hasta que se produzca la reparación del mismo).

Este modo de emergencia, que constituye un estado degradado del funcionamiento del sistema, debe contemplar el mantenimiento de un servicio lo más parecido posible al servicio no degradado (por ejemplo la regulación de un cruce por semáforos que se alternen por periodos de tiempo preestablecidos) o si ello no es posible, debe indicar inequívocamente que el sistema se ha degradado (por ejemplo, un semáforo en ámbar intermitente).

En el modo de emergencia ante un fallo en la comunicación (aislamiento de un dispositivo), el mismo debe estar preparado para restablecer el servicio cuando se restablece la comunicación, preferentemente de modo automático.

3.1.6. REQUERIMIENTOS DE FIABILIDAD POR APLICACIONES

Analicemos ahora algunas aplicaciones ejemplo en ITS que incorporan reguladores y dispositivos como los anteriormente expuestos, indicando en cada caso sus necesidades de fiabilidad en las comunicaciones.

3.1.6.1. Control de semáforos

Un cruce regulado por semáforos es uno de los casos más críticos respecto a los requisitos de fiabilidad de un dispositivo de señalización y de coordinación entre diferentes señales. De hecho, la regulación de cruces por medio de semáforos es muy anterior a los ITS basados en la moderna tecnología de las comunicaciones. En este caso, la simplicidad del funcionamiento de cada semáforo por separado no requiere un microprocesador en cada uno de ellos. El conjunto de semáforos que regulan un cruce están controlados por un autómata que controla directa y físicamente el funcionamiento de cada semáforo, asegurando que nunca se producirá una señalización con resultados fatales (en caso de detectarse algún problema, todos los semáforos del cruce señalarán luz ámbar). No se puede hablar en este nivel de comunicación de datos entre 'computadores'.

El control del conjunto de semáforos de una intersección sí se implementa a partir de un microprocesador de manera que puede implementar un protocolo de comunicación para recibir órdenes de cambio del ciclo y reparto de la intersección provenientes de un nodo superior, actuando sobre la red viaria. El controlador del cruce conoce el estado de funcionamiento (correcto o incorrecto) de cada uno de los semáforos, comunicándolo a los nodos superiores, especialmente en caso de avería. Por ello es importante la fiabilidad de la comunicación bidireccional del controlador con el nodo superior en la red de ITS.

Cuando un controlador de cruce no recibe comunicación (o la recibe incorrecta) puede mantener los parámetros en vigor durante un cierto tiempo, utilizar unos parámetros por defecto o señalar ámbar en todas direcciones, según la decisión del diseñador.

La armonización entre los controladores de cruces situados en una misma vía constituye un problema de señalización coordinada. Un regulador de zona establece comunicación con un conjunto de controladores de cruce. La comunicación con éstos no puede ser independiente dado que si ciertos cruces están coordinados y otros no, la situación creada no será crítica en cuanto a riesgo de accidentes pero sí producirá una congestión considerable. Sin embargo ninguna de los enlaces controlador de cruce-regulador de zona es más crítico que los demás ni justifica provisión de tolerancia a fallos más allá de la posibilidad de retransmisión de bloques dañados. En caso de fallo permanente de un enlace

(p.e. rotura de una línea, avería de un puerto de comunicación), el error debe ser detectado y el fallo reparado en el menor tiempo posible.

3.1.6.2. Carriles reversibles

El empleo de carriles reversibles supone también un problema de señalización coordinada que normalmente se resuelve de modo similar a los cruces semafóricos. Sin embargo, la señalización a lo largo de un carril reversible puede suponer una extensión mucho mayor que la correspondiente a un cruce semafórico. Por ello, en lugar del uso de líneas dedicadas exclusivamente al automatismo de la señalización del carril puede sustituirse por un control de la señalización a través de una red, siempre que se garantice la imposibilidad de señalizaciones incoherentes. Un fallo en la comunicación con una señal hace que ésta indique la situación de peligro (indicación de abandonar el carril).

La existencia de carriles reversibles sin alternativa (vías de un solo carril) hace que la disponibilidad de la señalización sea más crítica en términos de seguridad del sistema. La situación de indisponibilidad por la cual ninguno de los dos sentidos emplea el carril no es aceptable por lo cual se hace necesaria una redundancia mayor que permita que el sistema siga dando servicio no degradado incluso en presencia de fallos simples.

3.1.6.3. Paneles de Mensajes Variables

El control de un PMV se basa en varios elementos, cada uno de los cuales tiene su propia lógica interna. Cada elemento debe ser parametrizado según el entorno del panel y según las capacidades deseadas. En primer lugar el módulo de control, habitualmente situado al pie del panel, asegura la transformación de una orden exterior en una orden elemental del panel. Igualmente debe comprobar la correcta ejecución de la orden y conservar información de la traza de uso e incidencias del dispositivo. Normalmente, el módulo de control es suministrado conjuntamente con el propio panel.

El órgano de control, que puede ser local o estar situado en un Centro de Control de Tráfico, contiene la lógica necesaria para generar las órdenes. Los órganos de transmisión permiten hacer llegar las órdenes desde el órgano de control hasta los módulos de control de los paneles.

Los paneles deben ir provistos de mecanismos para comprobar a distancia tanto la correcta transmisión / recepción como la generalidad del funcionamiento interno del panel. Es pues conveniente guardar una traza tanto de los funcionamientos erróneos como de

todos los funcionamientos no ordenados por el órgano de control central, así como un histórico de las órdenes ejecutadas.

El panel debe tener un funcionamiento especificado para el caso de que el panel se encuentre en modo degradado o inactivo, que en cualquier caso no debe producir mensajes incoherentes con los de otros paneles activos.

La señalización presente en un panel tiene un periodo de validez determinado; al terminar dicho periodo se vuelve automáticamente al mensaje de reposo (también definido en las especificaciones). Dicho periodo de validez puede presentar diversas modalidades: en el caso de un sistema supervisado activamente, el periodo de validez puede ser ilimitado hasta que reciba una orden de cancelación; en el caso de un sistema con supervisión no permanente, el periodo se fija para una duración estimada, al cabo de la cual la señalización debe ser nuevamente confirmada o queda anulada.

La naturaleza de la señalización de algunos paneles (relacionada con su situación geográfica), por ejemplo la señalización de puerto de montaña cerrado, hace que sus requisitos de fiabilidad en términos de disponibilidad sean mayores. En caso de que la comunicación falle repetidamente (p.e. por un corte de la línea) y no se tenga confirmación de que la orden de señalización esté realmente activada en el lugar se hace necesario el desplazamiento físico de una persona al lugar en cuestión para emplazar la señalización. En este caso, la redundancia de la comunicación en ambos sentidos para tolerar los fallos está claramente justificada.

Un PMV puede formar parte de un **sistema de alerta autónoma** estando coordinado con un sistema de medida y/o un sistema de detección automática de incidentes (DAI). Cuando la medida sobrepasa un valor umbral o, en el caso más elaborado, cuando el DAI detecta un tipo determinado de incidente, el panel es activado con una orden adecuada (mostrar un determinado mensaje y/o pictograma).

Las comunicaciones dentro del sistema autónomo tienen unos requisitos de fiabilidad para que el panel no refleje alarmas inexistentes. Por su parte, el conjunto debe estar supervisado y verificado periódicamente a distancia desde un centro de control por el explotador del sistema.

En la gestión de una vía o conjunto de vías se utiliza un conjunto de paneles de mensajes variables que deben estar coordinados y gestionados por un órgano de control, constituyendo un ejemplo de señalización coordinada. Los paneles están coordinados por medio de las órdenes de un regulador, que puede limitarse a hacer cumplir los planes

recibidos desde órganos superiores, o constituirse como regulador inteligente en contacto con una estación de medidas, de modo que el regulador inteligente contiene una lógica que le permite activar y enviar órdenes adecuadas y coherentes al conjunto de PMVs. Cada panel puede ser controlado separadamente. El sistema central puede estar continuamente en contacto con el regulador inteligente con o sin validación de la activación, o simplemente ser verificado periódicamente.

Un **sistema integrado** es aquel que permite la integración de todas las funciones en un mismo sistema informático, utilizando sistemas de transmisión compatibles entre sí. Por ejemplo, es posible, a partir de un único nodo inteligente visualizar los flujos, controlar la rotación de una cámara, y enviar órdenes a un PMV. La integración puede ser completa si todas las transmisiones se hacen en una misma red.

Estos sistemas se encuentran especialmente en las vías de flujo intenso, donde el menor incidente puede tener consecuencias graves tanto sobre el plan de seguridad como sobre el tiempo perdido por los usuarios. Todo es posible, desde lo más sencillo hasta lo más complejo, cuando este caso está debidamente justificado por los objetivos que debe cumplir el sistema.

En cualquier caso, cuanto más modulares sean los procesos y más ajustados a los estándares sean los intercambios, la aplicación será más satisfactoria, en particular para asegurar las posibilidades de ampliación del sistema con nuevas funciones y su aplicación a áreas geográficas más extensas (además de no estar atado a un sólo proveedor-mantenedor sin cuya tutela técnica el sistema no sea viable).

3.1.6.4. Vigilancia de túneles

Existen aplicaciones de captación de datos y detección de incidentes, como puedan ser la vigilancia de túneles por medio de cámaras de televisión con detección automática de incidentes, donde la necesidad de disponer de datos provenientes de dicho subsistema es mayor que en otras aplicaciones de medición / captación /detección. Un fallo permanente o de larga duración por el motivo que sea en el enlace de comunicación supone que los intentos repetidos de transmisión fracasarán careciendo en consecuencia de información difícil de conseguir por otros medios de un punto que puede ser problemático para el tráfico. En tal caso habría que enviar personal al lugar provisto de medios de comunicación personal hasta que se restablezca la comunicación a través del enlace.

3.1.6.5. Procedimientos sancionadores automatizados

Los procedimientos sancionadores automatizados, juntamente al establecimiento de las garantías jurídicas adecuadas [Aur96] deben asegurar su funcionamiento y proporcionar la información necesaria para sustentar dichas garantías jurídicas. La detección de una infracción se acompaña actualmente de la captación de una imagen del vehículo infractor, para proceder a su identificación y procedimiento sancionador a partir de dicha imagen (***video enforcement***).

El proceso alcanza la completa automatización cuando la identificación y los procesos administrativos posteriores se realizan también de forma autónoma. Estos procedimientos sancionadores automatizados, de uso común en tramos con señalización fija (p.e. sanciones por exceso de velocidad controlada por radar), deben contemplar las características de los sistemas de señalización variable.

Los sistemas de señalización variable, regulados legislativamente a nivel internacional, permiten tanto señales informativas como señales coercitivas para la regulación de tráfico. Algunos ejemplos de estas señales reguladoras del tráfico son la señalización de acceso/prohibición a los carriles reversibles, la señalización de un carril BUS-VAO (carril reservado a Autobuses y Vehículos de Alta Ocupación, con dos o más ocupantes) y la señalización de límite de velocidad en un Panel de Mensajes Variables dentro de un sistema de control lineal de la velocidad, entre otros muchos.

La automatización de las sanciones que violen las indicaciones de señalización variable exige también la seguridad de que la restricción que afectase al vehículo infractor hubiese sido señalizada correctamente. Esto permite que las autoridades tengan la garantía suficiente de que los posibles infractores han hecho caso omiso de una señalización existente en un intervalo temporal conocido. Ello exige características especiales de fiabilidad y seguridad de estas aplicaciones que se tratan más extensamente en el capítulo sexto de esta memoria.

3.2. TOLERANCIA A FALLOS EN LA COMUNICACIÓN EN ITS

La fiabilidad en las comunicaciones en los ITS se consigue por la tolerancia de los fallos que inevitablemente se producirán. En los siguientes subapartados se exponen los medios

de corrección que se aplican ante la detección de errores en la transmisión, analizando las medidas de redundancia empleadas en cada caso.

3.2.1. FORMAS DE REDUNDANCIA

La tolerancia a fallos vendrá dada en cualquier caso por redundancia en las comunicaciones dentro de los ITS pudiendo ser, tal como se indicaba en el primer capítulo, de tres tipos:

3.2.1.1. REDUNDANCIA DE LA INFORMACIÓN:

La redundancia en la codificación permite la detección de errores y, eventualmente, la corrección de los mismos. La utilización de la redundancia en la información para la corrección, por codificación, del mensaje una vez recibido constituye la base de las técnicas **FEC** (*Forward Error Control - recovery*). En otros casos, la utilización de información redundante se limitará a la detección de codificaciones incorrectas que indica que el mensaje ha sido alterado durante la transmisión; en tal caso se utilizarán otros procedimientos para la corrección del error de transmisión, como las técnicas **ARQ** (*Automatic ReQuest for retransmission*). En ambos casos se debe incluir la cantidad suficiente de información redundante para permitir al receptor deducir cuándo se ha producido el error, y en el caso de las técnicas FEC, deducir cuál era el símbolo transmitido. La primera estrategia utilizará **códigos correctores de errores** (con mayor cantidad de información redundante), mientras que la segunda utiliza **códigos detectores de errores** [Rifa91].

La aplicación de técnicas de FEC (corrección de errores) exige un importante coste de cálculo, mucho mayor que el requerido para únicamente detectar los códigos incorrectos. Por ello, el uso de técnicas FEC queda casi siempre limitado a implementaciones hardware, dependiendo en cualquier caso de su necesidad en relación con la tasa de errores del medio de comunicación, el tipo de errores que puedan producirse y el tiempo permitido para su corrección. En aplicaciones de ITS, la redundancia de información para la detección es absolutamente necesaria, mientras que la redundancia para la corrección no está difundida, posiblemente debido a la falta de estandarización de los circuitos correctores de errores.

3.2.1.2. REDUNDANCIA TEMPORAL:

La utilización de la redundancia temporal es la base de las técnicas de ARQ. En caso de detectar que el mensaje recibido es erróneo, disponiendo el receptor de un canal de retorno, puede solicitar al transmisor una retransmisión de la información dañada. Igualmente el transmisor retransmite bloques de datos si sospecha que pueden no haber llegado a su destino (por no recibir confirmación de la recepción en un plazo determinado). Las técnicas de ARQ son más sencillas de implementar que las técnicas del tipo FEC traduciéndose esta sencillez en una mayor fiabilidad, mientras su desventajas principales son el uso de ancho de banda para la retransmisión y los retardos que se introducen. Las técnicas ARQ son pues desaconsejables en el caso de aplicaciones de tiempo real [Swe91].

Existen multitud de protocolos de comunicaciones estándar que usan técnicas ARQ. Sirva como ejemplo el protocolo de ISO HDLC, basado en el SDLC (*Synchronous Data Link Control*) de IBM y que incluye al X.25 LAPB del CCITT. Las tramas de HDLC incluyen un campo de control donde se indica el número de la trama que se está enviando y el número de la trama que se espera recibir. Además, pueden utilizarse tramas especiales de supervisión que pueden indicar el rechazo de una trama concreta. HDLC soporta dos modos de ARQ: GBN (*GoBackN*: repetición no selectiva) y SR (*SelectiveReject*: repetición selectiva) [Swe91].

La redundancia temporal (retransmisiones) puede ser gestionada a distintos niveles como se verá en el análisis por capas.

3.2.1.3. REDUNDANCIA ESPACIAL:

La redundancia espacial en las comunicaciones supone que en caso de fallo de un enlace de comunicación la información puede ir del punto origen a su punto destino por más de un camino. Esta posible redundancia espacial puede ser obtenida por dos formas diferentes:

una red de comunicación que disponga de alternativas de encaminamiento:

la comunicación entre un nodo y otro no conectados directamente puede hacerse a través de más de una ruta. Esta opción, siendo aplicable a las grandes redes con capacidades alternativas de encaminamiento, no es aplicable a los dispositivos simples de tráfico que transmiten su información directamente a un regulador y no tienen posibilidades de encaminamiento.

líneas redundantes:

tanto en el caso de línea punto a punto como en el caso de líneas de difusión, estas pueden estar replicadas (duplicadas en el caso más sencillo). Esto supone que la comunicación entre dos nodos conectados directamente puede hacerse por dos o más líneas físicas, suponiendo que cada nodo tiene un interfaz para cada una de estas líneas (o un sólo interfaz con posibilidad de cambiar de una a otra).

3.2.2. FIABILIDAD DE LAS TOPOLOGÍAS DE REDES LOCALES

Las posibilidades de comunicación en una red local dentro de una aplicación ITS dependen de la topología empleada, que a su vez está condicionada por la distribución geográfica de los nodos que deben comunicarse y por el coste total de la infraestructura de comunicaciones requerida. La topología elegida tendrá una fiabilidad respecto a sus posibilidades de tolerancia que se analiza a continuación, teniendo en cuenta el hecho de que las redes locales en ITS no se componen de un número de nodos similares que deben comunicarse entre sí, sino que la configuración más común es la de un nodo principal (regulador) que se comunica con cada uno de los otros.

Estrella

La configuración en estrella es la más adecuada para una red local en ITS dado que en ella el camino que debe recorrer cada bloque de datos es en principio mínimo (únicamente fluye entre el nodo principal y cada nodo terminal).

En esta topología, el fallo de un enlace es independiente del funcionamiento de los demás enlaces de la red local, por lo cual la comunicación entre el nodo regulador y los restantes nodos terminales queda intacta. Esto constituye un confinamiento de los errores que incrementa la fiabilidad global del sistema.

En las redes locales no se establecen caminos alternativos a los nodos terminales a través de otros terminales, ya que esto exigiría la utilización de protocolos que gestionasen el encaminamiento (y la utilización de un protocolo de nivel de red) que complica el funcionamiento de red y disminuye sus prestaciones de modo apreciable si se trata de enlaces lentos. Las comunicaciones directas entre un regulador y los restantes nodos de una red local en estrella son las más sencillas y robustas, no requiriendo servicios de encaminamiento.

Por tanto, una manera de tolerar un fallo en un enlace entre el regulador y un nodo terminal cuya comunicación se considere crítica es la duplicación de dicho enlace (la duplicación de cada enlace es independiente de la de los otros). Dicha duplicación no exige una capacidad adicional de encaminamiento ya que la comunicación se establece entre los mismos nodos.

El inconveniente principal de la configuración en estrella es la mayor longitud total de líneas requeridas, por lo cual se utilizan también las topologías de bus y anillo.

Anillo

La topología en anillo empleada en aplicaciones ITS para áreas periurbanas e interurbanas (normalmente implementada mediante fibra óptica) supone la utilización de un multiplexor-bifurcador para cada nodo terminal. Cada nodo terminal tiene capacidad de comunicarse independientemente con el regulador. Sin embargo esta capacidad de comunicaciones simultáneas no es un requisito absoluto del sistema.

La fiabilidad viene limitada por el hecho de que la falta de protección ante la rotura del anillo. Por contra la protección ante el mal funcionamiento de algún nodo se obtiene por la estructura de conexión de doble anillo (utilizada entre otros en los anillos FDDI) que permite que los datos fluyan hacia al regulador en uno u otro sentido, lo cual proporciona un cierto grado de tolerancia a fallos.

La estructura de anillo, especialmente empleando fibra óptica, supone una mayor complejidad técnica y especialmente un coste elevado de la instalación, resultando que en la mayoría de los casos la capacidad de comunicación proporcionada es infrautilizada. Por ello algunos sistemas ITS utilizan las topología de bus como solución de compromiso.

Bus

La topología de bus requiere la mitad de líneas que un anillo. Lleva implícita la utilización de un protocolo maestro-esclavo (estación primaria-estación secundaria) lo cual, como se ha dicho, no es un handicap en las aplicaciones ITS que se tratan.

El inconveniente de la topología de bus en cuanto a fiabilidad es que la protección contra la rotura del enlace o contra actuaciones negligentes es nula. Entre sus ventajas, destaca que esta topología supone el menor coste en equipos y líneas, a costa de una menor fiabilidad de la comunicación.

3.2.3. REDUNDANCIA DE ENLACES

La necesidad de tolerar fallos permanentes o fallos transitorios prolongados precisan la disposición de más de un canal entre dos computadores adyacentes (en el sentido de conexión directa entre ellos, independientemente de la distancia). La existencia de más de una línea directa de comunicación entre dos nodos adyacentes supone una redundancia espacial. La doble conexión entre nodos adyacentes es utilizada en desarrollos de amplia difusión como los anillos FDDI y los anillos autocurativos de las redes de transmisión Sonet/SDH [Tan96].

La gestión de la redundancia espacial de enlaces punto a punto ha sido tratada en telemática desde planteamientos 'propietarios' hasta intentos de estandarización que se describen a continuación.

3.2.3.1. MULTILINK PROCEDURES

Los niveles de enlace de X.25 y X.75 también soportan *multilink procedures*. Estos procedimientos permiten el uso de múltiples enlaces entre STEs (*Signaling Terminal Exchange*) estableciendo las reglas para la transmisión y sincronización a través de múltiples enlaces. Las operaciones multienlace permiten el uso de canales de comunicaciones paralelos entre STEs de manera que parezca un solo canal con capacidad mayor.

La operación multienlace también permite mayor fiabilidad que la proporcionada por un canal sencillo. Cuando se envían datos a través de un enlace simple con LAPB, si el enlace es defectuoso o produce demasiadas retransmisiones, se puede dirigir el trafico a otro enlace del grupo multienlace. También puede transmitir múltiples copias de un bloque de datos a través de mas de un enlace. La entidad receptora descartará las copias múltiples.

Los procedimientos multienlace se sitúan en la parte alta de la capa de enlace. La capa de red X.25 considera que está conectada a un solo enlace, mientras que los enlaces simples LAPB operan como si estuvieran conectados directamente a la capa de red. Los *multilink procedures* son responsables de la comunicación entre la capa de enlace y la capa de red [ISO 7478].

3.2.3.2. MULTILINK PPP

También se han definido procedimientos multienlace para el Protocolo de comunicaciones Punto a Punto (PPP) descrito en el capítulo cuarto. Estos procedimientos son similares a los descritos por ISO 7776.

La idea original de estos procedimientos era que el usuario dispusiera de ancho de banda adicional bajo demanda en enlaces RDSI. Dado que se permite la coordinación de múltiples enlaces, incluso de distinta naturaleza física, se pueden usar estos procedimientos para gestionar tolerancia a fallos mediante el uso de las líneas redundantes [Con95].

3.3. ANÁLISIS DE LA FIABILIDAD POR CAPAS EN REDES DE ITS

La organización de las redes de computadores en niveles o capas considerando cada una de ellas como un componente capaz de detectar errores provenientes de niveles inferiores e intentar subsanarlos antes de que transciendan a las capas superiores se adapta de manera adecuada a los modelos de componentes idealizados para tolerancia a fallos descritos en el capítulo primero. Se analizan en este apartado diversos aspectos de la fiabilidad de las redes empleadas en los ITS, en particular la contribución de cada uno de los niveles a la fiabilidad de las comunicaciones en ITS.

3.3.1. CAPA FÍSICA

La capa física se ocupa de la transmisión de bits a lo largo de un canal de comunicación, cuidando todos los detalles relativos al medio físico, con aspectos mecánicos, eléctricos y de procedimiento de interfaz con el medio. El nivel debe suministrar los medios necesarios a la activación, mantenimiento y desactivación de las conexiones física, resolviendo, entre otras, la elección de una codificación para cada una de las informaciones elementales (sea digital o analógica) y la elección de un modo de transmisión síncrono o asíncrono. Las principales normas a este nivel son la RS-232-C (CCITT V24) y sus sucesoras y la X24.

En los enlaces lentos con alta tasa de errores es preferible utilizar corrección de errores [Swe91] siempre que el tiempo empleado en la codificación-corrección sea menor que el tiempo necesario para las eventuales retransmisiones (la aplicación de técnicas FEC supone un tiempo adicional para todas las tramas, mientras las técnicas de ARQ sólo suponen

tiempo adicional ante tramas erróneas o perdidas). Debe tenerse siempre en cuenta que la técnica FEC es cara en términos de ciclos de CPU si debe hacerse por software y desproporcionada si los enlaces tienen una tasa de errores baja [Car96]

Los medios de transmisión, como GSM, propensos a errores y relativamente lentos (GSM funciona a 9600 bps), reducen globalmente el ancho de banda necesario usando técnicas FEC corrigiendo al valor del código más cercano. Dado que, como hemos apuntado, la implementación de técnicas FEC para corrección automática de errores debe hacerse por hardware, estas técnicas están siempre restringidas a la capa física, no empleándose nunca la corrección de errores por codificación en las capas superiores.

Sin embargo, las técnicas FEC no sirven para nada ante la pérdida de paquetes, en lugar de pérdida de bits, habiéndose comprobado que la mayoría de las grandes redes experimentan con mayor frecuencia la pérdida de paquetes que la alteración de bits [Car96].

El medio físico elegido para enlazar los diferentes nodos de la red ITS determina tanto la velocidad máxima de las transmisiones como la fiabilidad de las mismas, en conjunción con las condiciones externas que afectan a la red de comunicación.

En cualquier caso, los medios físicos más avanzados (por ejemplo fibra óptica) proporcionan velocidades de transmisión elevadas y tasas de errores muy bajas si se utiliza además un aislamiento adecuado. De hecho, para la mayoría de las aplicaciones de ITS supone una capacidad de transmisión muy superior a la necesaria. Sin embargo, incluso los medios más sofisticados no son inmunes a los fallos, y si el fallo es permanente la comunicación queda imposibilitada.

Las especiales características de las redes en ITS y su entorno sugieren que en ocasiones es más económico y preferible instalar enlaces redundantes mediante medios físicos menos sofisticados (más lentos y más propensos a errores). La redundancia de enlaces simples, que puede resultar menos costosa que un enlace de mayor capacidad infrautilizado, asegura además la tolerancia a fallos, incluso permanentes.

3.3.2. CAPA DE ENLACE DE DATOS

El objetivo de la capa de enlace es convertir un medio de transmisión no ideal (con ruido, retardos y colisiones) en un canal libre de errores de transmisión [Tan91], por tanto, queda clara su estrecha participación en la provisión de la tolerancia a fallos en la

comunicación directa (entre dos computadores conectados punto a punto o conectados a un mismo canal de difusión).

La capa de enlace tiene presentes las limitaciones de la línea física (ruido, retardos, velocidades de procesamiento finitas) y trata de evitar que ciertos errores transciendan a los niveles superiores. Para cumplir este objetivo, las funciones típicas de los protocolos de una capa de enlace genérica son: entramado, control de errores, control de flujo, y control del enlace [Sta94].

3.3.2.1. Tipos de servicios de la capa de enlace

El servicio que la capa de enlace ofrece a la capa de red puede ser de tres tipos distintos: no orientado a conexión, con o sin asentimiento, y orientado a conexión. En el servicio sin conexión y sin asentimiento, las capas de enlace de las entidades implicadas en la comunicación no establecen una conexión. Si debido a errores de transmisión, una trama es incorrecta o se pierde, no existe ningún método que permita recuperar el error. Este tipo de servicios están reservados para enlaces donde la tasa de errores es baja, y/o donde éstos van a ser recuperados por las capas superiores.

El servicio sin conexión y con asentimiento supone una mejora en la fiabilidad del enlace, pues aunque no se establece una conexión, se dispone de un método para indicar al transmisor cuando se produce un error. Mediante el uso de tramas especiales de asentimiento, el receptor valida la información recibida.

Por su parte, un servicio orientado a conexión garantizan que el enlace de datos recibe todas las tramas, sin la existencia de duplicados, y además en el orden correcto.

3.3.2.2. Entramado

La función de entramado consiste en componer la tramas (unidades de datos de protocolo de la capa de enlace), entregándoselas a la capa física para su transmisión. La trama la integra el paquete de datos proporcionado por la capa de red y campos de control añadidos por la capa de enlace.

Los campos de control se utilizan entre otras cosas para identificar el tipo de trama, indicar su procedencia y destino, indicar el número de secuencia, especificar la longitud de la trama, y para incluir información redundante que permita la detección y/o corrección de errores de transmisión [Tan91].

El entramado debe permitir además que el servicio de capa de enlace soporte diferentes protocolos de nivel superior. Esta, que es una de las características más destacadas del protocolo punto a punto PPP, es relevante en los ITS donde un enlace punto a punto a punto puede utilizarse tanto para la comunicación directa como para actuar como enlace parte de una transmisión extremo a extremo que implique el uso de protocolos de niveles superiores.

3.3.2.3. Control de errores

La función de control de errores de la capa de enlace debe determinar, haciendo uso de la información redundante, si se han producido cambios o pérdidas en los bits que componen la trama (cuando se usa hardware especial para la detección y corrección de errores, la capa de enlace no detecta errores, únicamente es informada de los errores no corregidos a nivel de capa física).

Se puede producir un error de transmisión cuando una o varias perturbaciones afectan a la señal que se transmite por el canal. Tales perturbaciones pueden ser distorsiones, atenuaciones, interferencias y ruido. También, puede ocurrir que se pierda la sincronización en recepción, de manera que el receptor sea incapaz de determinar el comienzo y fin de un bit, imposibilitando la detección de los valores de los bits que componen la trama. O incluso, debido al efecto paso bajo de todo canal, las transmisiones en banda base se encuentran afectadas por el fenómeno de Interferencia entre Símbolos.

Cabe recordar aquí, aunque sea una labor propia de la capa física, que la codificación de canal consiste en realizar un conjunto de transformaciones a la señal que representa la información a transmitir, para mejorar la eficiencia de la comunicación compensando los efectos negativos del canal de comunicaciones. La codificación de canal puede considerar dos aspectos: la forma de onda y las secuencias estructuradas, consistiendo estas últimas en el resultado de transformar las secuencias de datos originales en "mejores secuencias" para una eficiente detección y corrección de errores [Rifa91].

3.3.2.4. Control de flujo

Existen ocasiones en las cuales una entidad transmisora se ejecuta sobre un sistema rápido (o poco cargado), mientras que la entidad receptora lo hace sobre una máquina más lenta (o más cargada). Las tramas recibidas serán almacenadas en espera de que puedan ser procesadas, pero dado que toda capacidad de almacenamiento es limitada, llegará el instante en que el sistema receptor comenzará a perder tramas.

El control de flujo trata de evitar que esto se produzca, adaptando las capacidades de transmisión y recepción. En general, todo control de flujo va a exigir de algún mecanismo de realimentación que permita informar al transmisor cuando puede enviar una trama.

3.3.2.5. Control del enlace

El control del enlace se encarga de aquellas tareas de mantenimiento y administración del enlace establecido entre dos máquinas adyacentes. En los servicios sin conexión la administración es mínima, mientras que para los servicios orientados a conexión el control del enlace es bastante más complejo.

En los servicios orientados a conexión la forma de actuar consiste en tres fases: establecimiento de la conexión, transmisión de la información y desconexión. En la primera fase se debe solicitar y configurar el enlace, existiendo la posibilidad de negociar ciertas opciones. En la fase de transmisión, deben ejecutarse diversos procedimientos que permitan comprobar la calidad del enlace, y determinar si se produce una desconexión imprevista. Mientras que en la última fase se debe de indicar el deseo explícito de uno de los extremos de finalizar el enlace.

3.3.2.6. Protocolos ARQ en el nivel de enlace de datos

En la capa de enlace de datos se pueden utilizar dos familias básicas de protocolos denominados: ***stop-and-wait*** y **ventana deslizante** (*sliding window*) [Tan91].

En el protocolo *stop-and-wait*, el emisor toma un paquete del nivel de red, construye la trama, la envía por el enlace y se queda esperando a que se produzca un evento. El evento que espera es la llegada de una trama con el asentimiento (*acknowledgement*) de la trama previamente transmitida. En caso de no recibir el asentimiento en un tiempo fijado (*time out*) se procede a la retransmisión de la trama.

El protocolo *stop-and-wait* realiza simultáneamente un control de errores y un control de flujo. Si la trama que se recibe es incorrecta, no se envía el asentimiento, esperando que la retransmisión proporcione una trama libre de error. Mientras que el receptor no procese la trama y transmita el asentimiento, el transmisor está esperando (al menos el tiempo de *time out*).

Al funcionamiento básico de este protocolo se le pueden añadir técnicas adicionales, que en determinadas circunstancias, mejoraran la eficiencia del mismo. Ejemplos de estas técnicas son: la **superposición** y los **asentimientos negativos** (*Nak*). La técnica de superposición consiste en enviar el asentimiento de una trama recibida correctamente, en

un campo de control de la siguiente trama de información a transmitir. Así no se utilizan tramas especiales de control para enviar el asentimiento. Los asentimientos negativos se utilizan para que el receptor pueda indicar cuando ha recibido una trama errónea. De esta forma el transmisor no tiene que esperar el *time out* para retransmitir la trama.

En el protocolo de ventana deslizante el transmisor envía un número continuo de tramas sin esperar el asentimiento. Cada trama dispone de su propio *time out*, de tal forma que si no llega antes el asentimiento se produce la retransmisión. Se dispone de entidades llamadas ventanas de emisión y de recepción. La ventana de emisión representa las tramas enviadas de las que aún no se ha recibido el asentimiento. Mientras que la ventana de recepción se corresponde con las tramas que se pueden aceptar en cada momento.

Si la ventana de recepción tiene tamaño 1, la capa de enlace de datos sólo acepta las tramas en el orden correcto. Mientras que si la ventana es mayor que 1, cuando se reciba una trama errónea, se pueden guardar las posteriores de tal forma que cuando se retransmite la dañada, se dispone de una secuencia de tramas conectadas. Los protocolos con ventana de recepción de tamaño 1 se conocen con el nombre de ***Repetición no Selectiva*** (GBN, son las siglas de la denominación inglesa *Go Back N*). Dado que sólo admiten las tramas ordenadas, cuando una trama se recibe dañada, debe descartar todas las recibidas después y hasta que se produzca la retransmisión. Los protocolos con ventana de recepción mayor de 1 y que almacenan las tramas hasta disponer de una secuencia ordenada de tramas correctas, se denominan de **Repetición Selectiva.**

Se puede mejorar el protocolo añadiendo la posibilidad de asentimientos negativos (*Nak*), que son transmitidos cuando se recibe una trama incorrecta o cuando la trama recibida no era la esperada. Al recibir un *Nak* se retransmite únicamente la trama asociada, continuando con la trama con la que se interrumpió. No se retransmite una trama a menos que se indique explícitamente con un *Nak*, se evita con ello el generar duplicados innecesarios.

La conveniencia de uso de protocolos ARQ en el nivel de enlace de datos dependerá del tipo de protocolos utilizados por las otras capas. En las propuestas de estandarización para los ITS, particularmente en las del NTCIP, el protocolo de enlace de datos, sea PPP o PMPP, cumple simplemente la función de detección de errores mediante el usos de códigos cíclicos (CRCs) de tal modo que la detección de una trama incorrecta supone el descarte de la misma. Por tanto, la corrección por retransmisión se delega a los niveles superiores.

3.3.3. CAPA DE RED

La capa de red se ocupa del control de la gestión de una red. Se encargará de las funciones tendentes a la transferencia de información entre estaciones no contiguas. La función principal de esta capa es la determinación de las rutas que deben seguir los paquetes desde el computador que lo genera hasta el computador o computadores destinatarios de cada paquete, el llamado encaminamiento (*routing*); este problema es específico de las redes que se componen a su vez de redes más sencillas.

Otra de las funciones de la capa de red es el control de la congestión en la red. Se produce el fenómeno de congestión en una red cuando aumenta el tráfico en la red, de tal forma que los paquetes llegan a su destino con un retraso elevado, provocando la retransmisión de los mismos en su origen correspondiente, aumentando aún más el tráfico e incrementando el problema.

Otra función de la capa de red es la interconexión entre redes diferentes, cuando un paquete tiene que viajar de un computador de una red local hasta otra diferente, adaptándose a las especificaciones de cada una de ellas [Sta94].

Las rutas pueden obtenerse, según la red y sus principios de operación, tanto de modo estático a partir de tablas construidas según las conexiones existentes entre computadores de la misma, como de modo dinámico, considerando la carga en cada momento de cada tramo de la subred y encaminando los paquetes por las rutas menos congestionadas. Los algoritmos de encaminamiento, adaptativos o no, deben intentar garantizar:

- que los paquetes alcancen su destino en el menor tiempo posible,
- que los paquetes no deambulen por la red en caminos cerrados,
- que no se generen duplicados innecesarios que aumenten el tráfico y puedan inducir a error, y
- que las rutas defectuosas sean detectadas a tiempo y no supongan un "agujero negro" donde desaparezcan los paquetes [Tan91].

Uno de los algoritmos de encaminamiento más robustos es el de inundación, que garantiza que los paquetes llegan con total seguridad. Cuando llega un paquete se retransmite por todos las líneas, excepto por la que ha llegado. Evidentemente, debe limitarse el número de duplicados con algún método (contador de saltos, o número de secuencia), pues el tráfico de duplicados podría llegar a colapsar la red. El algoritmo

garantiza además que el paquete alcanza su destino por el camino más corto, pues selecciona todos los posibles en paralelo. Este algoritmo es óptimo para aplicaciones militares (donde en un instante determinado puede destruirse parte de la red).

La comunicación dentro de las redes locales de los ITS (ver figuras 2.10 y 2.11) no requiere servicios de encaminamiento, pues la comunicación se establece siempre entre el regulador y otro nodo. Por ello, la fiabilidad de las redes locales de los ITS se basará en la fiabilidad de la topología de la red local y en la fiabilidad proporcionada por los niveles inferiores y, en menor medida, de los mecanismos de tolerancia a fallos implementados en los niveles superiores.

3.3.3.1. FIABILIDAD EN LA RED SUPERIOR

La comunicación entre reguladores y nodos de nivel superior en ITS (sean centrales de zona o directamente el centro de control) implica el uso de nodos con mayores capacidades de comunicación y por tanto capaces de gestionar protocolos más complejos.

Esta comunicación por encima de la red local se establece sobre redes dedicadas o sobre redes abiertas. En el caso de topologías en árbol se utilizan en muchas ocasiones líneas adicionales que enlazan nodos de igual nivel para que en caso de que un enlace 'vertical' falle, la comunicación se encamine a través del enlace 'vertical' con un nodo adyacente y el enlace 'horizontal' entre nodos del mismo nivel.

La fiabilidad de las comunicaciones en redes públicas también implica la necesidad de procedimientos de autentificación de los extremos de la comunicación. La fiabilidad en términos de seguridad (*security*) de las comunicaciones de ITS en redes abiertas es un aspecto poco estudiado en ellos, implicando el uso de criptografía para asegurar la integridad y la confidencialidad de las informaciones transmitidas [Rifa95].

3.3.4. CAPA DE TRANSPORTE

Su función básica es proveer a las capas superiores de un medio fiable de transporte de la información de extremo a extremo independientemente del tipo de red o redes que ésta deba atravesar [Come94].

Las funciones de la capa de transporte son: la gestión de la conexión extremo a extremo, fragmentación y reensamblado de los mensajes proporcionados por los niveles inferiores, control de flujo extremo a extremo, control de errores extremo a extremo, recuperación de

caídas de la subred. Algunas de las funciones descritas coinciden con las desarrolladas por la capa de enlace de datos. En realidad, son muchos los mecanismos similares utilizados en ambas capas, pero la capa de enlace los aplica sobre la línea física entre dos estaciones contiguas, mientras que la capa de transporte los aplica a través de la subred.

El protocolo Internet de entrega fiable en el nivel de transporte se conoce como TCP . El algoritmo de recuperación de errores de TCP no tiene como único objetivo la corrección de errores en la transmisión (usando retransmisiones), sino que también proporciona el medio para que TCP trate las congestión en enlaces intermedios. El protocolo TCP asume que los retrasos de la red subyacente y el nivel de enlace de datos son debidos a la congestión de los enlaces y no a protocolos más complejos (p.e. los que implicaría retransmisiones en dicho nivel de enlace). Por tanto, los retrasos introducidos por retransmisiones en un nivel inferior serían mal interpretados por TCP, dado que una mayor varianza en la llegada de paquetes podría ser tomada como una pérdida de los mismos y solicitar su retransmisión, o en el peor de los casos 'confundir' al autómata del TCP, con la consiguiente pérdida neta de eficiencia de la red [Car96]

Otros protocolos de transporte utilizados en los ITS, como el UDP, no incluyen la solicitud de retransmisiones, limitándose a descartar los paquetes erróneos.

3.3.5. CAPAS SUPERIORES

Las cuatro capas inferiores del modelo OSI proporcionan los medios para el intercambio fiable de datos, sin embargo esto puede no ser suficiente para determinadas aplicaciones que requieren un diálogo controlado y estructurado, que, si bien podría ser llevado a cabo por las propias aplicaciones, es una función que el modelo establece como parte de la capa de sesión. Es pues la primera de las tres capas orientadas a los servicios de usuario, permitiendo establecer gestiones de diálogo y sincronización de intercambios (con establecimiento de puntos de recuperación para vueltas atrás).

La capa de presentación proporciona servicios relacionados con la representación y significado de la información, es decir, la capa de presentación se ocupa de la semántica de los datos intercambiados entre las aplicaciones. Las capas inferiores garantizan una transferencia ordenada de bits sin atender al significado de lo que se transmite.

Su funciones básicas se encargan de compatibilizar los sistemas de representación de las entidades que se comunican, de la compresión de la información, y de garantizar la seguridad y confidencialidad por medio de las técnicas criptográficas.

Por último, la función principal de la capa de aplicación es definir la interfaz entre el sistema telemático y los procesos de usuario (aplicaciones) que hacen uso de ese sistema telemático para comunicarse. Como se ha comentado en el capítulo anterior, estas tres capas constituyen en los ITS el nivel del transporte (*transportation*) y en los ITS suelen reducirse a una sola capa (la de aplicación) que engloba funciones de las tres.

Cuando el nivel de transporte no incluye la gestión de las retransmisiones, como en el caso de que se utilice protocolo de transporte UDP (o simplemente cuando el nivel de transporte es nulo) es el protocolo de nivel de aplicación o las propias aplicaciones las que determinan si los datos recibidos son correctos (después de haber sufrido una primera comprobación en el nivel de enlace de datos) y si necesita que dichos datos sean retransmitidos o no. Por ejemplo, en un sistema de captación de datos de tráfico que envía la información segundo a segundo, la perdida de un bloque de datos puede ser obviada ya que al segundo siguiente se va a recibir información más actual; este sería un caso en que la aplicación determinaría que la retransmisión no es necesaria. Por contra, si se trata del envío de una orden de cambio de señalización a un panel de mensajes variables, si la aplicación detecta que la orden no ha llegado (o no ha llegado correctamente) es necesaria la retransmisión.

3.3.6. FIABILIDAD EN LA CAPA DE ENLACE DE DATOS

La mayoría de las organizaciones asignan al nivel de enlace de datos la misión de asegurar la corrección de los datos entregados a las capas superiores [BJ87]. En caso de no poder asegurar la corrección de los datos, estos no son entregados a la capa superior. Las retransmisiones a nivel de enlace de datos como se ha descrito y se utiliza en LAPB [ISO 7776] entre otros no es la práctica más común, dejando al nivel de transporte las funciones de corrección de errores por retransmisión (por debajo del nivel de transporte, el nivel de red incorpora otro tipo de tolerancia por reconfiguración en los encaminamientos).

En principio ejecutar un protocolo que utilice retransmisiones por debajo de otro que también lo haga puede producir efectos indeseables sin aumentar realmente la fiabilidad de la transmisión. Sin embargo, si no se emplea TCP y los protocolos de los niveles superiores no realizan retransmisiones, entonces podría tener sentido confiarlas al nivel de enlace de datos como se hace en el protocolo HDLC y derivados.

También tiene sentido pleno la fiabilidad en el nivel de enlace de datos (por el uso de retransmisiones) cuando existe dependencia entre los datos como es el caso de los protocolos que utilizan compresión. Aun en este caso, el beneficio depende de que el coste de reiniciar el algoritmo de transmisión sea o no mayor que el coste de la retransmisión. [Rand96].

Igualmente debe tenerse en cuenta que el nivel de transporte trata de asegurar la comunicación de extremo a extremo. En el caso de dos nodos alejados, una comunicación de extremo a extremo implica el atravesar cierto número de enlaces intermedios. En tal caso, la existencia en el camino de un enlace especialmente propenso a errores puede provocar la pérdida de un paquete y en consecuencia la necesidad de retransmitirlo a través de todos los enlaces intermedios, aumentando de este modo el ancho de banda requerido en todos ellos. Este tipo de circunstancias puede hacer aconsejable incrementar la fiabilidad de un enlace determinado, contribuyendo de esta manera a la disminución del ancho de banda requerido en los otros enlaces (en caso de tratarse de un enlace simple, el incremento de fiabilidad introducido por las retransmisiones a nivel de enlace de datos permitiría que estas se produjeran únicamente en ese enlace y no fueran necesarias en otros).

Si el enlace es propenso a errores y los temporizadores del nivel de enlace de datos son menores que los de los niveles superiores (TCP), la mayor fiabilidad del nivel de enlace de datos beneficia al funcionamiento del nivel de transporte. Además si el problema es una alteración de bits en lugar de la pérdida de paquete (aunque estos sean más frecuentes) se solicita la retransmisión inmediatamente en lugar de esperar al temporizador de TCP. [Mic96]

Si las retransmisiones del nivel de enlace de datos son rápidas, estos pueden llegar al otro extremo correctamente antes de que expire el temporizador de TCP, con lo cual se ganaría en eficiencia. Esta propuesta depende de un análisis del tiempo de transmisión (*round-trip*) de cada paquete. De hecho, TCP estima los tiempos de transmisión y establece según estos los temporizadores [Car96]. En general, añadir técnicas de corrección es una buena o mala idea según la bondad del medio (tasa de errores). Si el enlace es propenso a errores, es preferible recuperar los errores a nivel de ese enlace para evitar perdida global del ancho de banda de extremo a extremo [Sch96].

La posible existencia de enlaces críticos que constituyan el único camino entre dos nodos hace que, a diferencia de los fallos transitorios que se pueden tolerar mediante

redundancia temporal, un fallo permanente en uno de estos enlaces críticos sólo pueda tolerarse mediante redundancia espacial en el enlace punto a punto.

También las configuraciones de enlaces múltiples entre dos nodos adyacentes comportándose como un único enlace lógico con un mayor ancho de banda disponible, propia de los multienlaces, pueden ser aprovechadas para soportar la tolerancia a fallos del enlace lógico.

La gestión de enlaces redundantes está, por su naturaleza, por debajo del nivel de red (que supone alternativas de encaminamiento) por lo que es misión del nivel de enlace de datos transmitir una información única que debe fluir entre dos nodos por medio de las líneas replicadas (pudiendo transmitir la misma información por todas las líneas o realizar otro tipo de distribución).

Esta gestión de enlaces redundantes constituye un subnivel superior dentro del nivel de enlace de datos, presentando "por debajo" una interfaz con el protocolo de enlace de datos estándar y una interfaz "por encima" para la capa inmediata superior (IP en la mayoría de los casos).

El estándar propuesto en esta memoria para la gestión redundante de un conjunto de enlaces punto a punto es el PPP multienlace [RFC1717] situado como un subnivel superior dentro del nivel de enlace de datos por encima del protocolo de enlace de datos PPP estandarizado en Internet para la comunicación punto a punto. La ubicación de este MLPPP en la pila de protocolos se muestras en la tabla de la figura 3.3.

niveles	protocolo
APLICACIÓN	protocolo de aplicación (para ITS) (STMP/otros)
PRESENTACIÓN	nulo
SESIÓN	nulo
TRANSPORTE	nulo/UDP/TCP
RED	nulo/IP
ENLACE DE DATOS	protocolo Multienlace (MLPPP)
	protocolo punto a punto (PPP)
FÍSICO	interfaz físico

Fig. 3.3. *Ubicación de MLPPP en la pila de protocolos*

3.4. CONCLUSIONES

La existencia de requisitos de enlaces fiables en los ITS nos lleva a proponer el uso de redundancia espacial, entendida en este ámbito como el uso de enlaces redundantes directos entre dos nodos, como solución a los fallos permanentes en enlaces sin alternativas de encaminamiento, siendo además una solución integrable en la arquitectura y los estándares propuestos para las comunicaciones en los ITS.

La gestión de enlaces redundantes entre dos nodos no está contemplada en ninguno de los estándares propuestos para la comunicación en los ITS. Ello implica que la utilización de enlaces redundantes para aplicaciones ITS críticas en sus necesidades de comunicaciones fiables no es compatible con la arquitectura y las normas propuestas, debiendo establecer una red paralela y no estandarizada para las comunicaciones críticas.

La gestión de enlaces redundantes debe por tanto estandarizarse, para lo cual proponemos utilizar en los ITS un estándar proveniente del área de las telecomunicaciones, donde, la gestión de un conjunto de enlaces entre dos nodos se sitúa dentro de las propuestas de estándares telemáticos como un subnivel superior del nivel de enlace de datos. El estándar investigado en esta memoria para permitir la integración en las redes de ITS de un grupo de enlaces redundantes como si fueran un único enlace es el PPP multienlace [RFC1717] situado por encima del protocolo de enlace de datos estandarizado (PPP) y que se presenta en el capítulo siguiente.

CAPÍTULO CUARTO

PROTOCOLO PUNTO A PUNTO Y EXTENSIONES PARA FIABILIDAD

Dado que el objeto del presente estudio se centra en las comunicaciones punto a punto en ITS, es necesario considerar las configuraciones típicas de este tipo de redes. Hay que destacar que la conexión punto a punto a través de enlace serie es el método más viejo para conectar dos computadores, y prácticamente todos los computadores la soportan (RS232). También se han utilizado enlaces punto a punto históricamente en las redes públicas de datos PDN, donde debido a la gran distancia de separación se hacia imprescindible el uso de líneas intermedias que conectasen entre sí las distintas estaciones.

Existen además otros usos frecuentes de las comunicaciones punto a punto. Dos computadores cualesquiera pueden requerir una conexión privada entre ellos, estableciendo para ello una comunicación punto a punto. Un usuario de una LAN puede desear acceder a la misma desde un computador distante, estableciendo una conexión punto a punto con la pasarela de acceso a la misma. Incluso dos LANs distantes pueden conectarse entre sí mediante un enlace punto a punto entre sus pasarelas respectivas. Obviamente, toda comunicación de datos entre dos computadores suficientemente distantes va implicar una conexión punto a punto, a través de medios conmutados o no, entre los dos computadores implicados en la comunicación, o bien, entre uno de ellos y la pasarela respectiva del otro.

Como se ha expresado, la topología de estrella es la más adecuada en términos de fiabilidad para la sección local de las redes de ITS, estableciéndose enlaces punto a punto entre el regulador y los nodos terminales (sensores o controladores). También la red de nivel superior, que conecta los reguladores con el computador central, se establece sobre enlaces punto a punto.

El presente capítulo analiza las características del Protocolo Punto a Punto PPP, propuesto como protocolo estándar para el nivel de enlace de datos tanto en comunicaciones punto a punto de propósito general en Internet como en comunicaciones punto a punto en los sistemas ITS que emplean los protocolos de la familia Internet (ver la

referencia al NTCIP en el apartado 2.4 de esta memoria), y sus extensiones, particularmente el PPP fiable y el PPP Multienlace (MLPPP) por sus potencialidades para implementar la tolerancia a fallos en la comunicación punto a punto para incrementar de este modo la fiabilidad de los enlaces en aplicaciones criticas.

4.1. REQUISITOS PARA UN PROTOCOLO PUNTO A PUNTO ESTÁNDAR

Llegados a este punto es necesario considerar dentro del panorama actual de las comunicaciones entre computadores, el hito histórico que ha supuesto el crecimiento explosivo de la Comunidad Internet. Aquella red de computadores que surgió en los años 60 como proyecto del DARPA norteamericano, se ha convertido en nuestros días en la red de redes, en el soporte para la integración de las múltiples redes de computadores dispersas que permite la universalización de las comunicaciones informáticas. Los usuarios de Internet se cuentan por millones en la actualidad, y se espera que para finales de siglo sean cientos de millones. Este crecimiento de la Comunidad Internet ha impulsado el desarrollo continuo de la tecnología de comunicaciones, tanto a nivel de hardware como de software (protocolos).

En un principio eran pocos los nodos conectados a Internet a través de enlaces punto a punto, la mayoría de ellos se conectaban a través de LANs o WANs. Una de las razones que impedía esto era la carencia de un protocolo estándar para las conexiones punto a punto en Internet. A pesar de que surgió un estándar de hecho, el protocolo SLIP, su definición era insuficiente para convertirse en un estándar real para la Comunidad Internet. Fue así como comenzaron los trabajos que concluyeron en la definición de un Protocolo para comunicaciones Punto a Punto (PPP).

Los requerimientos exigibles a un Protocolo de comunicaciones punto a punto para convertirse en un estándar Internet, definidos en [RFC1547], son los siguientes:

Simplicidad: debe ser un protocolo simple. La arquitectura Internet sitúa la complejidad en la capa de transporte (TCP), siendo la capa de red (IP) bastante simple, pues proporciona un servicio datagrama no fiable. Es por ello que la capa de enlace de datos no necesita proporcionar más capacidades que la capa de red: no son un requisito la corrección de errores, secuenciamiento o control de flujo. No quiere decirse que estas funciones no se utilicen, dependerán del caso concreto (por ejemplo: en un enlace ruidoso se podría utilizar LAPB), pero la clave del diseño es la simplicidad.

Transparencia: debe ser un protocolo transparente a los niveles superiores. Debe transportar los paquetes entregados por el nivel superior sin producir ninguna modificación.

Entramado: el receptor debe ser capaz de identificar correctamente el principio y final de cada trama, y dentro de ésta debe ser capaz de localizar los limites de cada octeto y de cada bit. Por ello el protocolo debe de proporcionar un entramado eficaz.

Eficiencia de Ancho de Banda: debe de hacer un uso lo más eficiente posible del ancho de banda disponible.

Eficiencia en el Procesado del Protocolo: las cabeceras introducidas por el protocolo deben ser simples para que sean rápida y eficazmente procesadas.

Multiplexado de Protocolos de la Capa de Red: el protocolo debe soportar el multiplexado de distintos protocolos de capas superiores. Es frecuente que en una misma red IP puedan coexistir otros protocolos tales como AppleTalk, DECnet, IPX, etc. Será necesario que el protocolo punto a punto incluya algún campo que indique el tipo de protocolo que está transportando.

Múltiples Protocolos de la Capa Física y de la Capa de Red: dada la existencia de muchos tipos de enlaces punto a punto (serie o paralelo, síncronos o asíncronos, de baja o alta velocidad, eléctricos u ópticos), el protocolo de comunicaciones punto a punto debe ser capaz de soportar diversidad de protocolos de la capa Física y de la Capa de Enlace de Datos, y no debe inhibir el uso de cualquier tipo de enlace.

Detección de Errores: el protocolo de comunicaciones punto a punto debe proporcionar algún tipo de detección básica de errores. Aunque la corrección de errores en los protocolos Internet se deja a la capa de transporte, el detectar en la capa de enlace de datos puede ser un mecanismo útil para no mermar el ancho de banda con la propagación de tramas corruptas.

Normalización de la Longitud Máxima del Paquete (MRU): el protocolo de comunicaciones punto a punto debe de establecer una longitud máxima (MRU) por defecto a los paquetes que debe transportar. Este valor por defecto puede ser susceptible de algún tipo de negociación. En el caso de que la capa de nivel superior intente transmitir un paquete de longitud superior a la MRU, será rechazada y producirá un error. Dadas las características del tráfico común en una LAN, el valor por defecto de la MRU debería ser de al menos 1500 bytes.

Medios conmutados y no conmutados: el protocolo de comunicaciones punto a punto debe ser capaz de soportar tanto enlaces conmutados (dinámicos) tales como una conexión a través de la Red Telefónica Básica o través de RDSI, como enlaces no conmutados (estáticos), como por ejemplo un cable RS232.

Simetría: se debe operar simétricamente para maximizar la flexibilidad, ambos extremos deben tener las misma jerarquía lógica en la comunicación. No se utilizarán reglas estáticas preasignadas (Maestro/Esclavo) que eliminan la simetría.

Vivacidad de la Conexión: el protocolo de comunicaciones punto a punto debe incluir un mecanismo que automáticamente determine cuando un enlace está funcionando adecuadamente y cuando no. Este mecanismo se habilitaría por defecto, aunque se debe permitir la negociación de su inhabilitación (en situaciones donde sea un coste elevado su uso, como en el caso de redes públicas donde se paga por la utilización de ancho de banda). Cuando esté habilitado, debe ser capaz de detectar variaciones en el enlace en un tiempo prudencial, puesto que un enlace caído que se sigue utilizando puede producir efectos desastrosos.

Detección de Bucles: el protocolo debe ser capaz de detectar automáticamente bucles en el enlace. Debe proveer de algún método que detecte fallos que produzcan que aquello que se recibe un extremo sea lo mismo que transmitió instantes antes.

Detección de Errores de Configuración: se deben detectar rápidamente conexiones punto a punto mal configuradas. Por ejemplo, pueden producirse errores al conectar los cables de una pasarela remota, es por ello que el protocolo debe tratar de identificar de forma simple el extremo remoto antes de declarar la conexión como válida.

Negociación de las Direcciones de Red: en ocasiones será necesario conocer las direcciones de red de ambos extremos del enlace antes de comenzar a intercambiar información (por ejemplo, en el acceso telefónico a una pasarela). Por tanto, el protocolo de comunicaciones punto a punto debe disponer de un algoritmo, tan simple como sea posible y garantizando su finalización en cualquier caso, que permita la negociación de las direcciones de red.

Negociación de la Compresión de Datos: en toda comunicación es interesante el uso de algoritmos de compresión de datos para optimizar el aprovechamiento del ancho de banda disponible. El protocolo debe proporcionar de un mecanismo que permita negociar el uso de estos algoritmos, permitiendo el uso de distintos algoritmos y asegurándose que ambos extremos utilizan el mismo algoritmo.

Negociación de Opciones y Extensiones: los cambios en la tecnología y en las demandas de los usuarios provocan continuas evoluciones y desarrollos. El protocolo de comunicaciones punto a punto debe permitir futuras ampliaciones y permitir a su vez la experimentación. Una solución para garantizar esto consistiría en definir un protocolo de negociación de opciones. Este protocolo se utilizaría para la negociación de opciones actuales como la dirección de red, técnicas de compresión de datos, MRU, etc. y permitir la ampliación y la negociación de un gran número de opciones futuras, además de permitir el uso de otro tipo de enlaces y técnicas de encapsulado.

Vistos los requerimientos para la definición de un estándar de protocolo de comunicaciones punto a punto (PPP), se hace necesario repasar el conjunto de protocolos punto a punto y capas de enlace de datos existentes y comprobar los requerimientos satisfechos por cada uno de ellos.

Dentro de los protocolos utilizados en Internet cabe destacar los siguientes::

Protocolos DCN Local-Network: en la RFC 891 se describe el formato de trama de la capa de enlace de datos usado por el sistema Fuzzball para enlaces: asíncronos, síncronos orientados a carácter, DDCMP. HDLC, ARPANET 1822, X.25 y Ethernet Cumple los requerimientos de simplicidad, transparencia, entramado y eficiencia, pero incumple otros muchos: asume que la capa de red es IP exclusivamente, no detecta errores, no determina la MRU, no discute la detección de bucle y fallos de configuración, y no existe la negociación de opciones.

Protocolos Thinwire: descritos en la RFC 914, discute el uso de enlaces de baja velocidad en Internet. Se proponen tres protocolos: Thinwire I, II y III. Los dos últimos requieren el uso de un protocolo fiable de enlace de datos, propuesto en el Apéndice D de la RFC. No es un protocolo complejo, aunque incluye detección y corrección de errores, por lo que tanto no puede ser calificado como simple. Considera un tamaño de paquete de 32 bytes, que lo hace ineficiente para paquetes de datos grandes, contempla sólo enlaces asíncronos, la capa de enlace fiable es redundante sobre enlaces LAPB, y además, no considera ningún mecanismo para la negociación de opciones y para futuras ampliaciones.

Protocolo de Transferencia Asíncrona Fiable (RATP): proporciona detección y corrección de errores, secuenciamiento y control de flujo a la conexión punto a punto. Está dirigido a enlaces RS232 *full-duplex* aunque puede ser utilizado para otro tipo de enlaces. RATP es un protocolo complejo, descrito en la RFC 916, que resuelve problemas no requeridos, mientras que olvida otros que si son requeridos.

Especialmente no soporta los mecanismos de negociación de opciones y ampliaciones.

Requisitos para Pasarelas Internet: en la sección 3 de la RFC 1009, se discute brevemente los requerimientos para la transmisión de datagramas IP sobre un diversos de enlaces punto a punto, incluyendo X.25 LAPB, enlaces síncronos con entramado HDLC, y el Protocolo de Entramado de Líneas Serie del MIT para enlaces asíncronos. Simplemente los menciona como posibles candidatos y no entra en profundidad sobre ellos.

Serial Line IP (SLIP): el protocolo IP sobre Línea Serie de Rick Adams debido a su popularidad casi se ha convertido en un estándar de hecho, pero no es un estándar real. El protocolo SLIP descrito en la RFC 1055, cumple los requerimientos de simplicidad, transparencia y eficiencia de ancho de banda. Pero únicamente da soporte al envío de paquetes sobre líneas serie asíncronas, no permite la simultaneidad de protocolos de capas superiores distintos, proporciona un único protocolo de entramado (siendo completamente redundante cuando se utilizan enlaces LAPB), no incluye ningún mecanismo de detección de errores, y no permite ningún tipo de negociación y capacidad de ampliación.

Los organismos internacionales para la estandarización de las comunicaciones han propuesto protocolos para el nivel de enlace de datos:

ISO 3309 - Estructura de Trama HDLC: es un protocolo de capa de enlace de datos que proporciona entramado de paquetes transmitidos sobre enlaces síncronos orientados a bit. Una secuencia especial (*flag*) marca el principio y el final de las tramas, y un mecanismo de inserción de bits permite que el campo de datos contenga la secuencia *flag*. Se detectan errores haciendo uso de una secuencia de verificación de trama (FCS) de 16 bits. Por si mismo, este protocolo no cumple muchos de los requerimientos: no proporciona multiplexado de protocolos , no normaliza la MRU, no detecta fallos de configuración, y no proporciona un mecanismo de negociación y extensión. Pero, dado su simplicidad y que está ampliamente aceptado puede ser un componente esencial sobre el que se construya un protocolo de comunicaciones punto a punto.

ISO 6256 - Procedimientos HDLC Balanceados: especifica una capa de enlace de datos con corrección de errores, secuenciamiento y control de flujo. ISO 6256 se construye sobre ISO 3309 y sobre ISO 4335 (Elementos de Procedimientos HDLC).

No aporta ninguna característica, de las calificadas como requeridas, que el propio ISO 3309. Lo nuevo de este protocolo es innecesario y además complejo.

X.25 del CCITT y X.25 LAPB: X.25 del CCITT especifica un protocolo de la capa de red sobre circuito virtual, con secuenciamiento, control de flujo y libre de errores. X.25 incluye una capa de enlace de datos denominada X.25 LAPB, basada en los protocolos ISO 3309, 6256 y 4335. Esta capa de enlace no aporta nada nuevo, desde el punto de vista de los requerimientos de un protocolo punto a punto, que los protocolos ISO.

Por último reflejaremos brevemente otros protocolos, en su mayoría productos comerciales:

Protocolos punto a punto de Cisco: las pasarelas Cisco soportan tanto enlaces asíncronos usando SLIP, como enlaces síncronos usando el entramado simple HDLC, X.25 LAPB o X.25 completo. Utilizan un protocolo que garantiza la vivacidad del enlace y da soporte a múltiples protocolos.

Protocolo de entramado PC/IP del MIT: este protocolo proporciona un mecanismo para la transmisión de datagramas IP sobre enlaces asíncronos. El protocolo sólo permite conexiones de computador a pasarela, se proporciona control de flujo y un rudimentario método de negociación de dirección. El protocolo no implementa: la detección de errores, la determinación del estado de la conexión, la detección de fallos y la negociación de opciones. Además sólo da soporte a enlaces asíncronos.

Protocolo punto a punto Proteon p4200: el *router* Proteon p4200 soporta la transmisión de paquetes sobre enlaces síncronos orientados a bit. Encapsula los paquetes en tramas HDLC, utiliza la secuencia de verificación de trama para detectar errores, pero la forma de numerar el campo tipo de Protocolo es propia, no ajustándose a ningún estándar. Utiliza un protocolo sencillo para determinar la vivacidad del enlace. No incluye ninguna forma de negociación de opciones.

Protocolo punto a punto XNS síncrono: fue diseñado cumpliendo los requerimientos de simplicidad, transparencia, eficiencia, entramado de paquetes, multiplexado de protocolos, detección de errores, normalización de MRU, simetría, medios conmutados y no conmutados, estado de la conexión, negociación de la dirección de red y ampliaciones futuras. Aunque no cumple los requerimientos de múltiples protocolos de capa de enlace, detección de fallos y negociación de la

compresión de datos. Además el multiplexado de protocolos de red se efectúa utilizando un campo de 8 bits, que es insuficiente.

En base a los requerimientos exigidos y teniendo presente el conjunto de protocolos existentes, en su momento, surge el Protocolo para Comunicaciones Punto a Punto (PPP). Definido en [RFC 1661], se ha convertido en el estándar para las comunicaciones punto a punto dentro de Internet. Es un protocolo diseñado inicialmente para comunicaciones *full-duplex* entre dos computadores extremos iguales (simétricos). Por defecto, utiliza un método de entramado similar al HDLC, asumiendo que la información llega ordenada, aunque permite la configuración de un enlace fiable haciendo uso del entramado de LAPB.

4.2. COMPONENTES DEL PROTOCOLO PUNTO A PUNTO (PPP)

PPP incluye tres componentes principales:

- Un método para encapsular los paquetes provenientes de la capa de red, admitiendo diversos protocolos en esta capa.
- Un protocolo de control del enlace (LCP), para establecer, configurar y comprobar la conexión del enlace de datos.
- Una familia de protocolos de control de red (NCP), que permite establecer y configurar diferentes protocolos del nivel de red.

La estructura de la trama utilizada por defecto es similar a la definida por HDLC (ISO 3309). la cabecera con la que se encapsula el paquete de la capa de red, requiere tan sólo 8 bytes, pudiéndose reducirse a 4 ó 2 bytes en entornos donde el ancho de banda es crítico. Este método de encapsulado por defecto utiliza campos simples, permitiendo implementaciones de alta velocidad. Además, se permite el multiplexado de diferentes protocolos de la capa de red simultáneos sobre el mismo enlace.

El protocolo de control del enlace (LCP) permite la configuración y finalización del enlace, garantiza que se alcance un acuerdo entre ambos extremos sobre las opciones del formato de la trama, permitiendo también la detección de bucles y otros fallos de configuración. Además proporciona otros utilidades opcionales, como la autentificación de

la identidad del otro extremo de la conexión y la determinación de cuando el enlace funciona apropiadamente o no.

El PPP dispone de un mecanismo de autoconfiguración a través de una negociación ampliable. Cada uno de los extremos del enlace describe sus capacidades y requerimientos, con la intención de alcanzar el acuerdo en los valores de configuración. Una implementación concreta de PPP fija los valores por defecto contemplando las configuraciones más comunes, pero el mecanismo de negociación permite el cambio de dichos valores para optimizar el enlace. Este mecanismo se implementa haciendo uso del protocolo LCP.

La familia de protocolos de control de red (NCP) son los encargados de atender las necesidades especificas requeridas por los respectivos protocolos de la capa red de la implementación concreta. Dada la particularidad de cada familia NCP, y por razones de extensión, no son objeto de esta memoria la descripción de dichos protocolos.

4.2.1. DESCRIPCIÓN DEL PROCEDIMIENTO PPP

El funcionamiento del protocolo PPP [RFC 1661] en líneas generales se podría describir de la siguiente manera: para establecer la comunicación sobre un enlace punto a punto cada extremo envía paquetes LCP para configurar y comprobar el enlace de datos. Después del establecimiento del enlace y el haber negociado, a través también de LCP, los servicios opcionales, se envían paquetes NCP que eligen y configuran uno o más protocolos de la capa de red. Cuando han sido configurados los protocolos de red, puede comenzar el intercambio de información entre ambos extremos del enlace. El enlace se cierra explícitamente mediante un paquete de LCP o NCP, o bien si ocurre algún evento externo (expira un temporizador , o interviene directamente el administrador de red).

En el procedimiento de establecimiento, mantenimiento y finalización del enlace punto a punto, el enlace PPP transita por distintas fases:

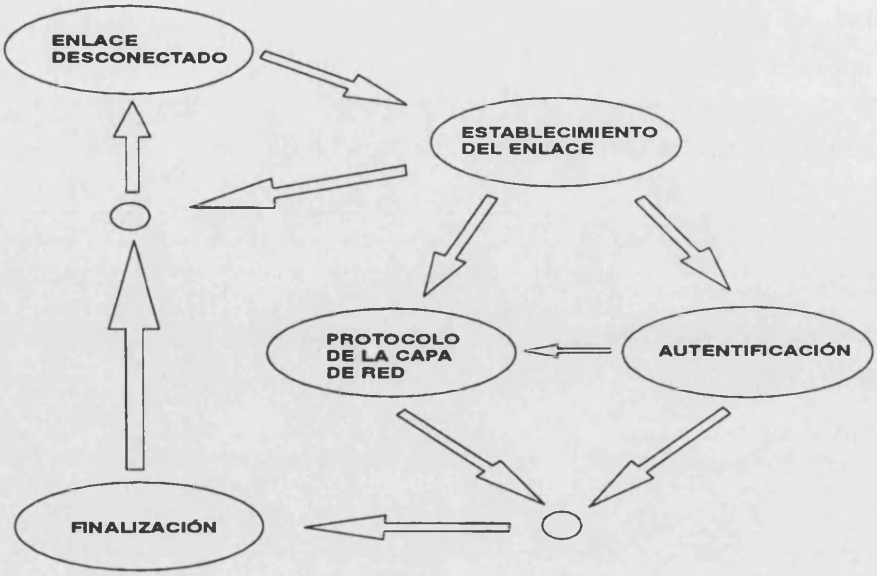


Fig. 4.1. *Diagrama de Fases del Enlace PPP.*

Enlace desconectado: es el estado inicial, y se permanece en esta fase hasta que un evento exterior indica que la capa física se encuentra preparada para la comunicación, lo cual da paso a la fase de establecimiento del enlace.

Establecimiento del enlace: la conexión se establece mediante el intercambio de tramas de configuración LCP. Los valores opcionales son configurados con los valores por defecto a menos que se indique lo contrario en el intercambio de tramas LCP. Sólo se pueden configurar mediante LCP aquellas opciones independientes de los protocolos concretos utilizados por la capa de red, siendo las opciones dependientes de éstos controladas por los protocolos NCP durante la fase de Protocolo de Capa de Red. Ninguna trama LCP recibida durante esta fase debe ser descartada "*silenciosamente*". Se completa esta fase cuando ambos extremos transmiten y reciben un paquete LCP *Configure-Ack.* Se retorna a esta fase desde las fases de Protocolo de Capa de Red y Autentificación, si se recibe un paquete LCP *Configure-Request.*

Autentificación: esta fase no es obligatoria, y se utiliza en aquellos casos que se requiera a un extremo a que se identifique antes de proceder al intercambio de paquetes del protocolo de la capa de red. En el caso de que una implementación quiera autentificar al extremo remoto con algún tipo de protocolo especifico, debe solicitar el uso de dicho protocolo de autentificación durante la fase de Establecimiento del enlace. En caso de que fallase la autentificación, el extremo autentificador pasaría a la fase de Finalización. La autentificación debe realizarse tan rápido como sea posible después de haber completado la

fase de Establecimiento, aunque puede permitirse que simultáneamente se ejecuten los mecanismos de determinación de la calidad del enlace, siempre que no retrasen indefinidamente la autentificación. Por ello, durante esta fase, los paquetes LCP recibidos distintos de la monitorización de la calidad del enlace y del protocolo de autentificación serán descartados "*silenciosamente*". Nunca se pasara a la fase de Protocolo de la Capa de Red sin haber completado con éxito esta fase.

Protocolo de la Capa de Red: finalizadas las fases previas, cada protocolo de la capa de red debe configurarse separadamente usando el apropiado Protocolo de Control de Red (NCP). Hasta que un NCP no haya activado su correspondiente protocolo de red no se aceptará ningún paquete de red de ese protocolo y aquellos que sean recibidos serán descartados "*silenciosamente*". Durante esta fase el tráfico en el enlace es una combinación de tramas LCP, NCP y tramas con paquetes de protocolos de la capa de red.

Finalización del enlace: PPP puede finalizar y cerrar un enlace en cualquier momento por diversos motivos: pérdida de la portadora, fallo de autentificación, baja calidad del enlace, expiración de un temporizador de inactividad, o por intervención del administrador de la red. El enlace se cierra mediante un intercambio de tramas LCP. Cuando el enlace se cierra, PPP informa a la capa de red para que ésta adopte las medidas oportunas. También se debe informar a la capa física para que proceda a la desconexión. Después de lo cual PPP pasaría a la fase de Enlace desconectado.

4.2.2. ENTRAMADO PPP

La especificación del entramado proporcionado por PPP [RFC 1662] soporta tanto enlaces síncronos orientados a bit y a octeto, como enlaces asíncronos con 8 bits de información sin paridad. Siendo los enlaces *full-duplex*, tanto conmutados como dedicados. Proporciona mecanismos de trasparencia y de detección de errores.

El entramado de PPP usa los principios de la Estructura de Trama HDLC descritos en ISO 3309, así como los procedimiento de control PPP hacen uso de la codificación del campo de control descrita en ISO 4335 (Elementos de Procedimientos HDLC).

La estructura de la trama tipo HDLC utilizada por PPP es la mostrada en la figura 4.2, y descrita a continuación.

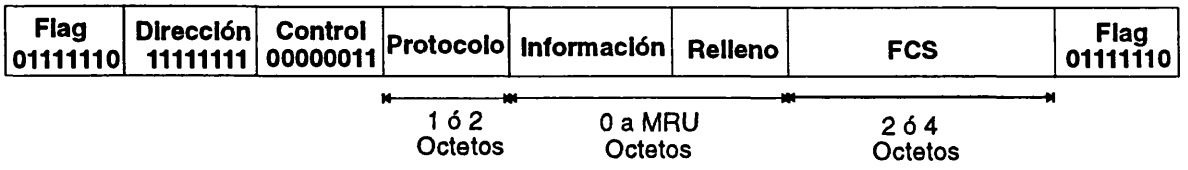


Fig. 4.2. *Estructura de la trama tipo HDLC.*

Secuencia de Inicio/Fin (*Flag*): todas las tramas comienzan y finalizan con la secuencia 01111110 (en hexadecimal 0x7E), se utiliza para la sincronización de la trama. Entre dos tramas consecutivas únicamente se requiere un *flag*. Dos *flag* consecutivos se consideraría como una trama vacía, siendo descartada "*silenciosamente*" (la trama es descartada sin producir ningún procesado extra).

Campo de Dirección: es un único octeto que contiene el valor 11111111 (0xFF), es decir, la dirección de todas las estaciones. El mecanismo de negociación puede acordar otras longitudes y valores del campo de dirección, pero siempre se debe poder reconocer la dirección 0xFF. Las tramas recibidas con una dirección desconocida serán descartadas "*silenciosamente*".

Campo de Control: contiene un sólo octeto con el valor 00000011 (0x03), es decir, indica que se trata de una trama de información no numerada. El uso de otros valores para este campo se puede acordar mediante el mecanismo de negociación.

Campo de Protocolo: identifica el protocolo de la capa de red cuyo paquete ha sido entramado. Su longitud puede ser uno o dos octetos, y su valor tiene que ser tal que el bit menos significativo del octeto menos significativo sea 1, mientras que el bit menos significativo del bit más significativo sea 0.

Campo de Información: el campo de información contiene el paquete proporcionado por la capa de red, y puede tener una longitud entre 0 y MRU octetos. El valor por defecto de MRU, incluyendo el campo de relleno, es de 1500 octetos, aunque mediante el mecanismo de negociación se puede acordar el uso de otro valor.

Campo de Relleno: en transmisión, al campo de Información se le pueden añadir octetos, con el limite máximo para el total de octetos de información y relleno del valor especificado como MRU. Es responsabilidad de cada implementación distinguir entre los octetos de información y de relleno.

Secuencia de Verificación de Trama (FCS, *Frame Check Sequence*): por defecto la longitud de este campo es de 2 octetos, aunque se puede negociar el uso de 4 octetos. El valor de la FCS se calcula sobre todos los bits de los campos de Dirección, Control, Protocolo, Información y Relleno, sin incluir los bit de *start* y *stop* (caso asíncrono), los bits insertados por razones de transparencia, y el *flag* de inicio/fin de trama. El polinomio generador para el caso de una FCS de 16 bits sería $x+x^5+x^{12}+x^{16}$, mientras que para el caso de 32 bits: $x+x^1+x^2+x^4+x^5+x^7+x^8+x^{10}+x^{11}+x^{12}+x^{16}+x^{22}+x^{23}+x^{26}+x^{32}$.

El entramado debe garantizar, con técnicas de transparencia, que la secuencia de inicio/fin de trama no aparece en los restantes campos de la misma. Esto se consigue haciendo uso del procedimiento de Inserción (*stuffing*). Según sea el enlace, se utilizará inserción de octetos o de bits.

4.2.2.1. INSERCIÓN DE OCTETOS

Se utiliza con enlaces síncronos orientados a octeto y enlaces asíncronos con un bit de *start*, 8 bits de datos y un bit de stop.

Se define el octeto de Escape como el valor 0x7D. El transmisor debe garantizar, como mínimo, la transparencia a la secuencia de inicio/fin de trama y al octeto de Escape. Aunque para el caso de enlaces asíncronos se puede negociar, mediante el uso de paquetes LCP, los caracteres de control para los cuales se debe garantizar transparencia. Este conjunto de caracteres de control transparente se denomina Mapa de Caracteres de Control Asíncrono (ACCM), y cada extremo del enlace debe mantener dos ACCMs, uno para transmisión y otro para recepción.

Antes de transmitir una trama, y después de calcular la FCS, se comprueba la trama completa. Si se detecta la secuencia de inicio/fin, el octeto de Escape, o cualquier otro octeto perteneciente al ACCM, se reemplaza por dos octetos: primero el octeto de Escape, seguido del resultado de aplicar la operación XOR entre el octeto original y el octeto 0x20.

En recepción, antes de calcular la FCS, se eliminan todos los octetos de Escape previo computo de la operación XOR del octeto siguiente al Escape con el valor 0x20.

Se consideran tramas no válidas, aquellas cuya longitud es pequeña (menos de 4 octetos cuando se usa una FCS de 16 bits), o aquellas que finalizan con un octeto de Escape seguido del *flag* de fin de trama, o en la cuales no se respeta el formato de un carácter (por

ejemplo, transmitiendo un bit 0 como bit de *stop* cuando se espera un 1). Las tramas no válidas son descartadas "*silenciosamente*".

4.2.2.2. INSERCIÓN DE BITS

Este procedimiento se utiliza en enlaces asíncronos orientados a bit. Se garantiza que en la trama no se producirá la secuencia de inicio/fin de trama.

Después del cálculo de la FCS, el transmisor examina la trama incluyendo todos sus campos entre el *flag* de Inicio y el de Fin, e inserta un bit 0 a continuación de cada secuencia de 5 bits 1 consecutivos (incluyendo los últimos 5 bits de la FCS). Por supuesto, en recepción se eliminarán todos los bits 0 precedidos de una secuencia de 5 bits 1.

Son descartadas "*silenciosamente*" aquellas tramas que bien son demasiado pequeñas o que finalizan con una secuencia de más de 6 bits 1.

4.2.3. PROTOCOLO DE CONTROL DEL ENLACE (LCP)

Mediante el Protocolo de Control del Enlace (LCP) se configura el enlace, negociando aquellas opciones más óptimas para el entorno concreto; se monitoriza, detectando bucles y fallos de configuración; y se proporciona mecanismos para cerrarlo cuando la comunicación finaliza.

Estas funciones se implementan mediante el intercambio de los oportunos paquetes LCP entre ambos extremos del enlace. Los paquetes LCP se describen a continuación. Por razones de extensión, en la presente memoria no se detalla el autómata de estados finitos que define el funcionamiento del LCP, éste se puede encontrar en [RFC 1661].

4.2.3.1. PAQUETES LCP

Los datos del protocolo LCP son encapsulados en el campo de información de las tramas PPP, indicándose en el campo de tipo de protocolo de la trama el valor 0xC021 (protocolo LCP). Desde este punto de vista los datos de este protocolo reciben el mismo

tratamiento que un paquete de la capa de red. Por ello se adopta la nomenclatura de paquete LCP para los datos de una trama LCP.

Existen tres tipos distintos de paquetes LCP:

- Paquetes de Configuración del Enlace: usados para establecer y configurar el enlace (*Configure-Request, Configure-Ack, Configure-Nak y Configure-Reject*).
- Paquetes de Finalización del Enlace: usados para cerrar el enlace (*Terminate-Request y Terminate-Ack*).
- Paquetes de Mantenimiento del Enlace: usados para la gestión del enlace (*Code-Reject, Protocol-Reject, Echo-Request, Echo-Reply, Discard-Request, Identification* y *Time-Remaining*).

Los paquetes LCP tienen un formato básico de 4 campos. Los campos básicos de los paquetes LCP son:

Fig. 4.3. *Formato de los paquetes LCP.*

Código: identifica el tipo de paquete LCP. tiene una longitud de un octeto. Los valores de este campo son los siguientes:

Código	Paquete LCP
1	*Configure-Request*
2	*Configure-Ack*
3	*Configure-Nak*
4	*Configure-Reject*
5	*Terminate-Request*
6	*Terminate-Ack*
7	*Code-Reject*
8	*Protocol-Reject*
9	*Echo-Request*
10	*Echo-Reply*
11	*Discard-Request*
12	*Identification*
13	*Time-Remaining*

En caso de recibir un paquete que contenga un valor desconocido en el campo Código, se transmitirá un paquete *Code-Reject.*

Identificador: es un campo de un octeto utilizado como auxiliar en algunos paquetes de solicitud y respuesta. Cuando se recibe un paquete con un Identificador no válido, el paquete debe ser descartado "*silenciosamente*".

Longitud: es un campo de dos octetos que indica la longitud del paquete LCP, incluyendo todos los campos (Código, Identificador, Longitud y Datos). La longitud máxima del paquete se haya limitada por la MRU. Si se recibe un paquete con una Longitud no válida es descartado "*silenciosamente*". Los octetos del paquete que superan la longitud indicada por este campo son considerados de relleno.

Datos: campo de longitud variable y cuyo formato viene determinado por el campo Código del paquete.

4.2.3.1.1. Paquetes de configuración del enlace

Se envía el paquete *Configure-Request* al extremo remoto para indicar que se desea establecer una conexión. Las opciones de configuración que el transmisor desea negociar se incluyen en el campo de datos del paquete. Todas las opciones son negociadas simultáneamente, y deben ser incluidas en el mismo paquete. En las opciones negociables no se debe indicar el valor por defecto.

El campo Identificador de este paquete no varía en caso de retransmisión. Sólo cambia su valor cuando se producen cambios en las opciones de configuración, o cuando se realiza una solicitud distinta.

En respuesta a un paquete de *Configure-Request,* el extremo remoto envía un paquete *Configure-Ack,* cuando todas las opciones de configuración recibidas son conocidas y sus valores aceptados. En tal caso, el valor del campo Identificador del paquete tiene que ser el mismo que el del paquete *Configure-Request* previamente recibido, y el campo de Datos debe contener exactamente la misma lista de opciones de configuración, para las cuales se envía el asentimiento. Los paquetes calificados como no válidos, por discrepancias en el campo Identificador o en la lista de opciones de configuración, deben ser descartados "*silenciosamente*".

Cuando los valores de las opciones de configuración no son aceptados, o bien cuando la implementación desea negociar una opción que no ha sido incluida en la lista del *Configure-Request*, el extremo remoto transmite un paquete *Configure-Nak*. El campo de Datos contiene la lista de las opciones que son asentidas negativamente. Sólo se incluirán las opciones cuyos valores no sean aceptados. Las opciones asentidas negativamente deben de indicar el valor aceptable por la implementación, pudiendo indicar el valor por defecto de la opción. Cuando se desea negociar una opción no incluida en el *Request*, se añadirá al paquete de *Nak* indicando el valor o valores aceptables para esa opción.

En recepción, un paquete *Configure-Nak*, debe tener el mismo Identificador que el último paquete *Configure-Request* enviado. Los paquetes no válidos serán descartados "*silenciosamente*". La recepción de un paquete válido *Configure-Nak*, supone el envío de un nuevo paquete de *Configure-Request*, donde las opciones de configuración han sido modificadas de acuerdo a lo indicado en el paquete de *Nak*.

El extremo remoto envía el paquete *Configure-Reject* cuando alguna de las opciones recibidas en la petición *Configure-Request*, no es reconocida o bien no se permite su negociación. El campo de datos de este paquete indica la lista de opciones que son rechazadas, no incluye las opciones reconocibles y negociables presente en el paquete de *Request*. La recepción de un paquete *Configure-Reject* supone el envío de un nuevo *Configure-Request*, donde la lista de opciones de configuración no debe incluir las opciones rechazadas.

4.2.3.1.2. Paquetes de finalización del enlace

Mediante la transmisión de estos paquetes, LCP proporciona un mecanismo para cerrar el enlace. Cuando un extremo desea finalizar transmite un paquete *Terminate-Request*, que debe ser contestado por el extremo remoto con una paquete *Terminate-Ack*. La transmisión de *Terminate-Request* debe continuar hasta que se reciba el asentimiento, o bien que la capa física indique la desconexión, o tras un número considerable de retransmisiones. El formato de estos paquetes incluye los campos de Código, Identificador, Longitud y Datos.

4.2.3.1.3. Paquetes de Mantenimiento del enlace

Los paquetes LCP utilizados para el mantenimiento del enlace son: *Code-Reject, Protocol-Reject, Echo-Request, Echo-Reply, Discard-Request, Identification* y *Time-Remaining.*

En caso de que el extremo remoto opere con una versión distinta de LCP, se pueden recibir paquetes LCP cuyo campo de Código no sea reconocible. Cuando ocurra tal circunstancia, se debe informar al extremo remoto transmitiéndole un paquete *Code-Reject.* La recepción de un paquete *Code-Reject* rechazando un paquete LCP fundamental para la versión utilizada por la implementación, supone el abandono de la conexión, dado que es improbable que la situación se pueda rectificar automáticamente.

Por otro lado, en caso de que el campo de Protocolo de la trama PPP recibida no sea reconocido, se envía el paquete *Protocol-Reject* al extremo remoto . Esto puede ocurrir habitualmente, cuando el extremo remoto intenta configurar un protocolo nuevo. En caso de recibir un paquete de *Protocol-Reject,* la implementación debe cesar, tan pronto como se tenga oportunidad, de enviar paquetes del protocolo rechazado.

Mediante el uso de los paquetes *Echo-Request* y *Echo-Reply,* LCP proporciona un mecanismo de interacción que permite ejecutar una transmisión en ambas direcciones del enlace. Esto es especialmente útil en situaciones tales como monitorización del enlace, determinación de la calidad del mismo, comprobación de prestaciones, y otras muchas. Estos paquetes incluyen un campo de 4 octetos denominado *Magic-Number,* utilizado para la detección de bucles (aquello que se recibe es lo que previamente se ha transmitido) en el enlace.

En esta misma línea, el uso de paquetes *Discard-Request* proporciona un mecanismo que permite ejecutar una transmisión desde el extremo local hacia el extremo remoto.

Por último el paquete *Identification* permite a una implementación identificarse a su extremo, y el paquete *Time-Remaining* permite notificar al extremo el tiempo restante de la conexión actual, siendo esta indicación de naturaleza únicamente informativa.

4.2.4. OPCIONES DE CONFIGURACIÓN

Las opciones de configuración LCP permiten negociar la modificación de las características por defecto del enlace punto a punto.

Una implementación que solicita una determinada opción, está indicando bien la disponibilidad o el requerimiento de opciones adicionales. Siempre que una opción de configuración no se incluya en un paquete *Configure-Request* se asume el valor por defecto de la misma. Los valores por defecto no es necesario que sean enviados en el paquete *Configure-Request.* El definir un valor por defecto para cada una de la opciones garantiza el correcto funcionamiento del enlace sin necesidad de negociar opciones, aunque posiblemente admitiendo una pérdida de prestaciones.

El formato básico de las opciones de configuración incluye tres campos: Tipo, Longitud y Datos:

Fig. 4.4. *Formato de las opciones de configuración.*

Tipo: es un octeto que indica el tipo de opción de configuración. Los valores para este campo serían:

0	Reservado
1	Unidad de Recepción Máxima
3	Protocolo de Autentificación
4	Protocolo de Calidad
5	*Magic-Number*
7	Compresión del campo de Protocolo
8	Compresión del campo de Control y Dirección
9	Alternativas FCS
10	*Relleno Self-Describing*
11	Transmisiones Fiables
13	Rellamada
15	Tramas compuestas

Longitud: es un octeto que indica el tamaño de la opción de configuración con todos sus campos, incluido el de Longitud. Si se recibe una opción de configuración con una Longitud errónea, se debe transmitir un paquete *Configure-Nak* que incluya esa opción con la Longitud y los Datos correctos.

Datos: de longitud variable, contiene la información específica de la opción de configuración. El formato y la longitud concreta de este campo viene determinado por el campo Tipo.

4.2.4.1. UNIDAD DE RECEPCIÓN MÁXIMA (MRU)

Se utiliza esta opción para informar al extremo remoto la posibilidad de recibir paquetes grandes, o para solicitar el envío de paquetes más pequeños. El valor por defecto de la MRU es de 1500 octetos, y cualquier implementación debe estar capacitada para recibir paquetes de este tamaño en caso de pérdida de sincronización del enlace y habiendo negociado un tamaño menor.

4.2.4.2. PROTOCOLO DE AUTENTIFICACIÓN

En algunos tipos de enlaces puede desearse el requerir a un extremo a autentificarse antes de permitir el intercambio de paquetes de protocolos de la capa de red. Esta opción de configuración proporciona un método para negociar el uso de un protocolo especifico para la autentificación. Por defecto no se requiere esta opción.

La implementación no debe incluir en la opción una lista de múltiples protocolos de autentificación. Debe solicitar el protocolo preferible, y sólo en el caso de que esa opción esté incluida en un paquete *Configure-Nak* del extremo remoto, se solicitará el uso del siguiente (en orden de preferencia) protocolo de autentificación.

Cuando un extremo solicita mediante un *Configure-Request* la autentificación del extremo remoto, y éste último acepta mediante un *Configure-Ack*, el extremo local esperará la autentificación del remoto haciendo uso del protocolo acordado.

4.2.4.3. PROTOCOLO DE CALIDAD

Mediante esta opción se negocia el uso de un protocolo especifico para monitorizar la calidad del enlace. Monitorizando la calidad se puede determinar cuando y con qué

frecuencia el enlace pierde información. Por defecto la monitorización del enlace no se encuentra habilitada.

No es obligatorio que la monitorización sea *full-duplex* ni tampoco que se use el mismo protocolo en ambas direcciones. El protocolo LQP propuesto para monitorizar enlaces PPP se describe en [RFC1333].

4.2.4.4. MAGIC-NUMBER

La opción de configuración *Magic-Number* proporciona un método para detectar bucles en el enlace (los datos que se están recibiendo son aquellos que la misma estación ha transmitido previamente). Por defecto, está opción no se negocia por lo que hay que insertar un cero allá donde se requiera un *Magic-Number*. En caso de negociar el uso de esta opción, se debe escoger el *Magic-Number* del extremo de la forma más aleatoria posible.

Cuando se recibe un *Configure-Request* con la opción de configuración *Magic-Number*, el valor recibido se compara con el valor transmitido en el último paquete *Configure-Request*. Si ambos valores son distintos, entonces no existe un bucle en el enlace y el *Magic-Number* debe ser asentido. Si ambos números son iguales, es posible que exista un bucle en el enlace y que el *Configure-Request* recibido sea el enviado previamente. Para comprobar la existencia real o no del bucle, el extremo local transmite un paquete *Configure-Nak* con un *Magic-Number* distinto.

La recepción de un paquete *Configure-Nak* con un *Magic-Number* distinto del que se transmitió al enlace en el *Configure-Nak* previo, indica la no presencia de bucles en el enlace. Si por el contrario los valores del *Magic-Number* vuelvan a coincidir, se incrementa la probabilidad de que exista un anomalía en el enlace. En tal caso se debe escoger un nuevo *Magic-Number* y volver a enviar un nuevo paquete *Configure-Request*.

En caso de bucle en el enlace, esta secuencia (transmisión de *Configure-Request*, recepción de *Configure-Request*, transmisión de *Configure-Nak*, recepción de *Configure-Nak*) se repetirá constantemente. Si no existe un bucle en el enlace, es tremendamente improbable que la secuencia se repita en muchas ocasiones, pues los valores de *Magic-Number* escogidos en ambos extremos rápidamente van a diverger. La tabla siguiente muestra la probabilidad de colisión, suponiendo que ambos extremos eligen el valor de *Magic-Number* con una distribución uniforme [RFC 1661]:

Número de Colisiones	Probabilidad
1	2.3 E-10
2	5.4 E-20
3	1.3 E-29

El *Magic-Number* también puede ser utilizado para detectar bucles durante el modo de operación normal. Los paquetes *Echo-Request*, *Echo-Reply* y *Discard-Request* disponen de un campo de *Magic-Number*, donde se transmitirá el valor previamente negociado. En recepción se comprobará que estos valores coinciden con el negociado para el extremo local. El recibir un paquete con el valor de *Magic-Number* del extremo local indicará un bucle del enlace.

4.2.4.5. COMPRESIÓN DEL CAMPO DE PROTOCOLO

Mediante esta opción de configuración se informa al extremo remoto que puede recibir campos de Protocolo, de las tramas PPP, con tamaño de 1 octeto. Por defecto el tamaño de este campo es de 2 octetos, pero sus valores son escogidos de tal manera que algunos pueden ser comprimidos en un sólo octeto.

Hacer notar que el Tipo de Protocolo se codificaba de tal forma que un 0 en el bit menos significativo indicaba que el siguiente octeto pertenece al campo de Protocolo, y un 1 en el bit menos significativo identifica al octeto como el último del campo de Protocolo. Así, para valores inferiores a 256, basta con utilizar un sólo octeto pues el primer octeto tendría valor 0.

La compresión del Campo de Protocolo es especialmente interesante en enlace de baja velocidad. No se utiliza esta compresión a menos que se haya negociado, y cuando ha sido negociada cualquier implementación debe aceptar tramas con ambos formatos. Las tramas que contienen paquetes de LCP nunca son comprimidas.

El cálculo del campo FCS de la trama se efectúa sobre la trama comprimida.

4.2.4.6. COMPRESIÓN DEL CAMPO DE CONTROL Y DIRECCIÓN

Usualmente el campo de Control y Dirección de las tramas PPP tienen valores constantes, por lo que pueden ser eliminados de la trama, aumentando con ello la eficiencia de ancho de banda. Mediante esta opción se puede negociar la compresión de los campos de Control y Dirección, pero por defecto está opción no se encontrará habilitada. Una vez negociada y aceptada la compresión, la implementación debe ser capaz de recibir tanto tramas comprimidas como no comprimidas. Las tramas que contienen paquetes de LCP nunca deben ser comprimidas. El valor del campo FCS debe ser calculado sobre la trama comprimida.

4.2.4.7. ALTERNATIVAS FCS

Esta opción de configuración permite especificar a una implementación un formato de FCS, y negociarlo con su extremo. Esta opción se puede negociar separadamente en cada dirección del enlace. Los valores de FCS negociados sólo se utilizarán durante las fases de Autentificación y Protocolo de la Capa de Red. Las tramas enviadas en cualquier otra fase del enlace utilizaran la FCS por defecto.

4.2.4.8. RELLENO SELF-DESCRIBING

Esta opción de configuración permite a una implementación informar a su extremo que es capaz de utilizar rellenos del tipo *Self-Describing*. Cada octeto del relleno *Self-Describing* contiene el índice de ese octeto. El primer octeto de relleno contiene el valor 1, el siguiente 2, y así sucesivamente. Después de quitar de la trama la FCS, el octeto final del relleno indica el número de octetos de relleno a eliminar.

En esta opción se negocia el Valor Máximo de Relleno (MPV *Maximum Padding Value*). Con lo que en el relleno sólo se usan los valores comprendidos entre 1 y MPV. Cuando no se requiere el uso del relleno, pero el último octeto del campo de información de la trama PPP contiene un valor comprendido entre 1 y MPV, al menos se debe añadir un octeto de relleno a la trama. Si el octeto final es mayor que MPV no hace falta añadir ningún relleno.

4.2.4.9. RELLAMADA

Esta opción de configuración proporciona a la implementación un método para solicitar al extremo remoto, conectado a través de una línea telefónica básica, que cierren la conexión y que la vuelvan a establecer, pero esta vez efectuando la llamada telefónica el extremo remoto. La opción de rellamada (*callback*) puede utilizarse para diversos propósitos, pero principalmente para ahorrar coste telefónico cuando un sentido es más caro que el otro.

Si la opción es negociada con éxito y se completa la fase de Autentificación, se pasa directamente de esta fase a la fase de Finalización del enlace, produciéndose la desconexión. Posteriormente se restablece el enlace, sin negociación de rellamada.

4.2.4.10. TRAMAS COMPUESTAS

Mediante esta opción de configuración se puede negociar el que una implementación pueda enviar múltiples paquetes PPP en una misma trama. Cuando se usan estas tramas, inmediatamente después de un paquete se añade el campo de protocolo del siguiente paquete.

Si es necesario añadir octetos de relleno al campo de Información, se debe utilizar relleno del tipo *Self-Describing*. Por tanto, esta opción se debe negociar junto con la opción del relleno *Self-Describing*.

4.2.5. TRANSMISIONES FIABLES PPP (MODO NUMERADO)

Los enlaces PPP, con el entramado HDLC que se ha descrito, proporcionan un servicio datagrama sin conexión y sin asentimientos. En aquellas circunstancias en que sea necesario disponer de enlaces más fiables [RFC 1663], donde no se pierda información y ésta pueda ser perfectamente reordenada, se puede utilizar un método de entramado tipo LAPB [ISO 7776] para modo Numerado (se trata de un protocolo de ventana deslizante que utiliza asentimientos positivos y negativos, así como técnicas de superposición). El uso del modo Numerado se debe especificar como opción de configuración en la fase de Establecimiento del Enlace.

El formato de la trama respecto al modelo descrito con anterioridad para el modo No Numerado (tipo HDLC), únicamente varía en el campo de dirección y en el campo de control.

El formato de la opción de configuración para el modo Numerado, incluye los campos de Tipo, Longitud, Ventana y Dirección. El campo Ventana es un octeto cuyo valor indica el tamaño de la ventana deslizante, es decir, cuántas tramas van a ser mantenidas en el *buffer* de recepción, que a su vez es el número máximo de tramas enviadas por el transmisor sin un asentimiento. Cuando el valor indicado es menor que 8 se usa el secuenciamiento denominado módulo 8, en otro caso se utiliza el módulo 128. Con el campo Dirección cada extremo elige y negocia su dirección. Se consideran legales todos las direcciones consistentes con ISO 3309.

El establecimiento del enlace se realiza de la forma habitual. Son negociadas las opciones de configuración de modo Numerado y *Magic-Number*, pero no se negocia la opción de compresión del campo de control y dirección. Aceptado el modo Numerado, éste debe ser utilizado en todas las tramas, puesto que algunas implementaciones no permiten el uso del campo de control No Numerado o el uso de la dirección 0xFF (todas las estaciones) en el modo Numerado.

Finalizada la negociación de la fase de Establecimiento del Enlace, el extremo que tenga el valor de *Magic-Number* más pequeño debe enviar una trama de SABM (establecimiento de la conexión) a lo que contestará el otro extremo con una trama UA (indica que se ha cumplido la orden anterior). Siguiendo a continuación con las restantes fases de enlace PPP descritas con anterioridad.

Los parámetros configurables del modo Numerado son los siguientes:

- Temporizador T1: tiempo máximo permitido antes de una retransmisión, como consecuencia de no recibir respuesta a la transmisión previa de una trama.

- Temporizador T3: da una indicación sobre el estado de inactividad del enlace. Su valor debe ser mayor que el valor de T1.

- Número máximo de intentos para completar una transmisión N2: es el valor máximo de retransmisiones de una trama concreta. Si se supera este valor el enlace se debe cerrar.

4.3. PROCEDIMIENTOS PPP MULTIENLACE

El protocolo PPP Multienlace (descrito en [I-D MP12]) es una extensión cuya motivación inicial es la posibilidad de establecer varios enlaces PPP simultáneos entre dos sistemas, dando a los usuarios un ancho de banda adicional bajo demanda. El objetivo del protocolo es la coordinación de un grupo (*bundle*) de enlaces independientes (si es necesario, incluso de distinta naturaleza física) entre un par de sistemas, proporcionando un enlace punto a punto virtual con mayor ancho de banda.

Los métodos del Multienlace PPP son similares al protocolo multienlace descrito en ISO 7776, añadiendo ciertas capacidades adicionales. No se requiere que la capa de enlace opere con asentimientos, permitiéndose como opción.

PPP Multienlace utiliza negociación de una opción LCP que permite a un sistema informar a su extremo de su capacidad multienlace:

- El sistema que ofrece la opción es capaz de combinar múltiples enlaces físicos dentro de un único enlace lógico.
- El sistema es capaz de recibir unidades de datos de protocolo (PDU) de la capa superior fragmentadas usando la cabecera multienlace, y es capaz de volverlas a ensamblar en la PDU original.
- El sistema es capaz de recibir PDU de tamaño N, incluso siendo N mayor que la MRU de un enlace simple.

Para modelar los procedimientos Multienlace, se puede considerar una entidad virtual de la capa de enlace donde los paquetes recibidos a través diferentes entidades de la capa física son identificados como pertenecientes a un protocolo de red PPP distinto (el Protocolo Multienlace, o MLPPP), siendo recombinados y secuenciados de acuerdo a la información presente en la cabecera de fragmentación multienlace. Todos los paquetes recibidos por enlaces identificados como pertenecientes al conjunto multienlace son entregados a la capa de red, tanto si tienen cabecera multienlace o no.

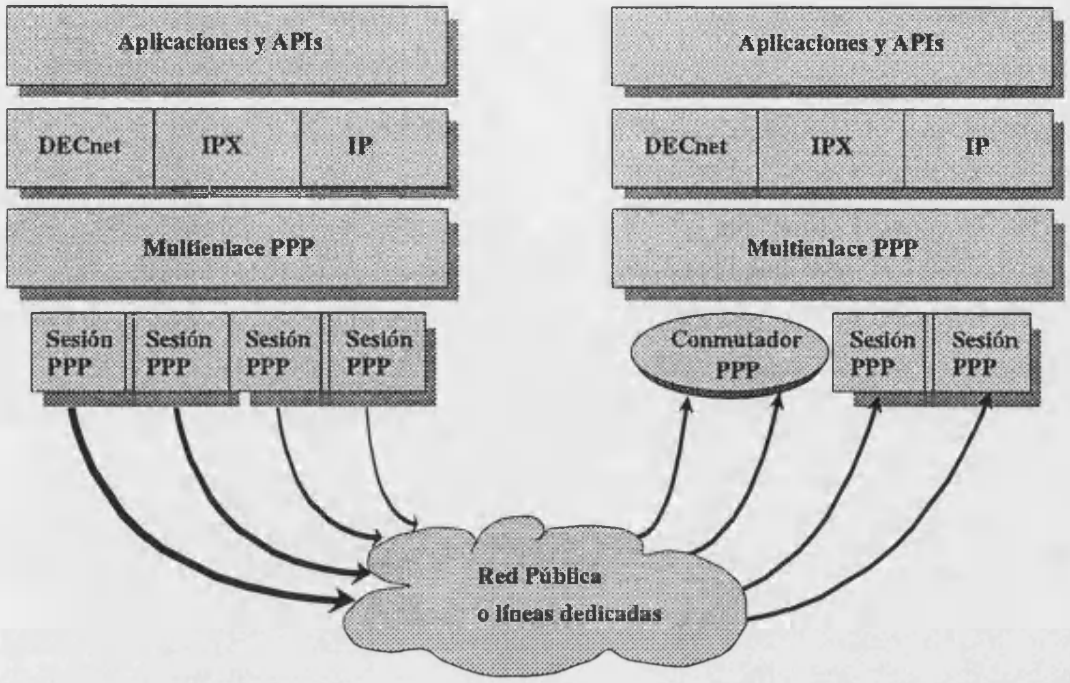


Fig. 4.5: *PPP Multienlace*

La negociación de las opciones de configuración se realiza de forma independiente e individual para cada enlace simple del conjunto. No se permite la negociación LCP sobre el conjunto multienlace. La implementación no debe transmitir paquete *Configure-Request*, *-Reject*, *-Ack*, *-Nak*, *Terminate-Request* o *-Ack* a través de los procedimientos multienlace, y si una implementación los recibe debe descartarlos "*silenciosamente*". Por contra, se permite la transmisión de otros paquetes con funciones de control no asociados al cambio de la configuración del conjunto multienlace. Se permite la transmisión vía procedimientos multienlace de los paquetes *Code-Reject, Protocol-Reject, Echo-Request, Echo-Reply* y *Discard-Request.*

La MRU efectiva para la entidad de enlace lógico es negociada vía opción LCP.

4.3.1. FORMATO DE LOS TRAMAS MULTIENLACE.

Los paquetes entregados por la capa de red son en primer lugar segmentados, si es necesario, para que tengan el tamaño apropiado para los procedimientos multienlace. A

continuación se le añade el Campo de Protocolo, completando así el encapsulado (no entramado) de los fragmentos de los paquetes de la capa de red.

A cada uno de los fragmentos encapsulados, el Protocolo Multienlace le añade dos campos nuevos: el Identificador de Protocolo Multienlace, y la Cabecera Multienlace. El campo Identificador de Protocolo son dos octetos que contienen el valor 0x003D. Mientras que la Cabecera Multienlace son cuatro octetos que contienen información para el secuenciado del paquete fragmentado. El primer octeto contiene en sus posiciones más significativas los bits B y E. El bit B con valor 1 indica el primer fragmento del paquete PPP, mientras que el bit E con valor 1 indica el último fragmento del paquete. Los tres octetos restantes contienen el número de secuencia del fragmento.

Mediante la negociación de una opción LCP adicional, se puede reducir el tamaño de la Cabecera Multienlace, pasando de cuatro octetos a dos. En tal caso se utilizan 12 bits para indicar el número de secuencia.

Las implementaciones multienlace pueden fragmentar los paquetes originales con distintos tamaños. En el caso de disponer de enlaces de distinta naturaleza, una estrategia posible consistiría en dividir los paquetes en tamaños proporcionales a la velocidad de transmisión de cada enlace, o bien dividir los paquetes en fragmentos iguales y distribuirlos con mayor carga sobre los enlaces más rápidos.

Para terminar de componer la trama multienlace, bastaría añadir los campos de Dirección y Control adecuadamente comprimidos, y añadir la secuencia de verificación de la trama (FCS) calculada para ese fragmento concreto. Las transmisiones multienlace sólo utilizan, en caso necesario, relleno del tipo *Self-Describing*, y sólo el fragmento final puede ser rellenado.

4.3.2. DETECCIÓN DE FRAGMENTOS PERDIDOS.

El transmisor envía fragmentos con número de secuencia estrictamente crecientes. La implementación inicia el número de secuencia con 0 en el primer fragmento transmitido para un multienlace recién establecido.

El receptor comprueba los números de secuencia que llegan en cada uno de los enlaces que componen el conjunto y mantiene el mínimo de los números de secuencia recibidos recientemente sobre todos los miembros del conjunto multienlace (este valor se denomina M). El receptor detecta el final de un paquete cuando recibe el fragmento que contiene el

bit E puesto a 1. Entonces puede reensamblar el paquete completo si se han recibido todos los paquetes con números de secuencia creciente hasta el último fragmento.

Se detecta la pérdida de un fragmento cuando el valor M supera el número de secuencia de un fragmento con el bit E a 1 de un paquete que no ha sido completamente reensamblado (no se han recibido todos los fragmentos con número de secuencia comprendida entre el paquete con el bit B a 1 y el paquete con el bit E a 1).

La detección de un fragmento perdido con número de secuencia U, produce que el receptor descarte todos los fragmentos recibidos con número de secuencia superior al fragmento final con número más bajo mayor o igual a U.

La regla de secuencia creciente prohibe el recolocar los fragmentos de la cola de transmisión de un enlace caído en la de un enlace activo. Esta práctica es usual en implementaciones ISO Multienlace sobre LAPB.

4.3.3. EXTENSIONES LCP PARA MULTIENLACE

El protocolo Multienlace PPP introduce nuevas opciones de configuración:

- Máxima Unidad Multienlace Reconstruida (MRRU).
- Discriminador del extremo.
- Formato de Cabecera Multienlace con Número de Secuencia Corto.

4.3.3.1. MRRU MULTIENLACE

La presencia de esta opción de configuración indica que el sistema que la envía implementa el protocolo PPP Multienlace. Un sistema debe incluir esta opción siempre que intente iniciar o unirse a un conjunto multienlace. No se envían paquetes multienlace hasta que la opción MRRU ha sido ofrecida y aceptada en la última negociación LCP. Si la opción no es ofrecida, el extremo puede incluirla en el paquete *Configure-Nak.*

Mediante esta opción se negocia el tamaño de la Unidad Reconstruida Recibida Máxima (MRRU), que corresponde con el número máximo de octetos de los campos de Información los paquetes reensamblados.

4.3.3.2. DISCRIMINADOR DEL EXTREMO REMOTO

Mediante esta opción de configuración, no obligatoria para los procedimientos multienlace, se puede identificar al extremo remoto. La opción avisa al sistema cuando el extremo del enlace puede coincidir con el extremo de otro enlace ya existente. Si se diferencia este extremo de todos los restantes, se establece un nuevo conjunto multienlace a partir de su negociación. Si los campos de Clase y Dirección (incluidos en esta opción) coinciden con alguno de los extremos de un enlace existente, el nuevo enlace se añade al conjunto multienlace que contiene el enlace con el extremo coincidente, o bien se establece un nuevo conjunto multienlace. Esto último depende de la posibilidad de uso de un Protocolo de Autentificación y del resultado del mismo.

4.3.3.3. FORMATO DE CABECERA CON NÚMERO DE SECUENCIA CORTO

Esta opción permite solicitar el uso de número de secuencias de 12 bits. Si la opción es aceptada se utiliza siempre este formato para todos los fragmentos enviados a través del multienlace.

4.3.4. FINALIZACIÓN DE ENLACES MIEMBROS DEL CONJUNTO MULTIENLACE

Los enlaces miembros se cierran de la forma habitual de los procedimientos PPP, usando paquetes LCP *Terminate-Request* y *Terminate-Ack*. Por supuesto, la recepción de un paquete *Terminate-Request* sobre uno de los enlaces no concluye el procedimiento en los enlaces restantes.

Mientras alguno de los enlaces miembros del conjunto multienlace permanezca activo, el estado MLPPP del conjunto persiste como una entidad separada. En caso de que quede un único enlace, si todos los demás enlaces han sido adecuadamente cerrados (con *Terminate-Ack*), la implementación puede cesar de usar la cabecera multienlace.

4.4. CONCLUSIONES

La utilización del protocolo punto a punto PPP se muestra como una opción idónea para su utilización en el nivel de enlace de datos de los enlaces punto a punto existentes en las redes de ITS por sus características:

- se trata de un protocolo Internet con lo que ello supone en cuanto a interoperatividad con otros nodos Internet;
- es un protocolo simple y robusto;
- hace un uso relativamente eficiente del ancho de banda, suficiente en la mayoría de aplicaciones de ITS;
- permite el multiplexado de protocolos de la capa de red;
- se puede establecer sobre múltiples protocolos de la capa física;
- incluye la detección de errores por CRCs implementada por software
- permite la negociación de opciones y extensiones.

Además, el protocolo PPP es un protocolo abierto a la negociación de opciones, entre las cuales resultan de interés los protocolos de autentificación y los protocolos de calidad. Los primeros garantizan la inviolabilidad de los enlaces a través de redes abiertas, lo cual impide la intromisión en las comunicaciones de las aplicaciones de ITS. Los protocolos de calidad permiten monitorizar los enlaces para tomar las medidas de mantenimiento y reparación adecuadas, aspecto éste no desdeñable en las redes de los ITS donde los enlaces están expuestos a problemas de campo y donde la extensión geográfica de la red hace aconsejable la monitorización remota de los enlaces.

En situaciones en que se necesite disponer de corrección de errores por retransmisiones en el nivel de enlace de datos, tal como se ha discutido en el apartado 3.3.6, PPP permite la utilización de la opción PPP fiable donde ante fallos transitorios no se pierda información y ésta pueda ser perfectamente reordenada.

Por su parte, la gestión de enlaces redundantes puede ser llevada a cabo por el protocolo MLPPP, que se basa en enlaces PPP simples implementables sobre casi cualquier protocolo de nivel físico. La transmisión a través de un multienlace permite la comunicación a través del mismo mientras exista al menos uno de los enlaces en funcionamiento correcto. Este cualidad, combinada con una adecuada monitorización del multienlace, permite que ante un fallo permanente de uno de los enlaces en una red de ITS el problema sea detectado para proceder a la reparación del mismo, mientras el servicio de enlace de datos sigue disponible.

CAPÍTULO QUINTO

PROPUESTA DE PROCEDIMIENTOS DE GESTIÓN DE LA REDUNDANCIA EN COMUNICACIÓN PUNTO A PUNTO

El presente capítulo está dedicado a presentar la aplicación que hemos llevado a cabo de los modelos de soporte para la tolerancia a fallos expuestos en el capítulo primero a la comunicación punto a punto para la gestión de la redundancia al objeto de obtener **servicio de enlace de datos punto a punto con tolerancia a fallos transparente**, siendo dicho servicio soportado por un conjunto de protocolos de nivel de enlace de datos, protocolos de nivel físico y hardware de comunicaciones, y entendiendo por **transparencia** la cualidad de que el servicio proporcionado sea externamente idéntico a un sistema normal sin tolerancia mientras no se produzca una avería, de modo que el resto del sistema se relaciona con el mismo igual que lo haría con uno no tolerante. Para los distintos modelos para tolerancia a fallos describiremos los procedimientos de tratamiento de la redundancia con un análisis de las ventajas e inconvenientes de cada uno de ellos.

5.1. MODELO DE COMPONENTE T.F.: SERVICIO DE ENLACE DE DATOS PUNTO A PUNTO TOLERANTE A FALLOS

Utilizaremos el modelo de componente tolerante a fallos descrito en el capítulo primero que se refleja en la figura 5.1 para expresar los diferentes elementos y actividades que sustentan un servicio de enlace de datos punto a punto asimilándolos con los diferentes items que constituyen dicho modelo.

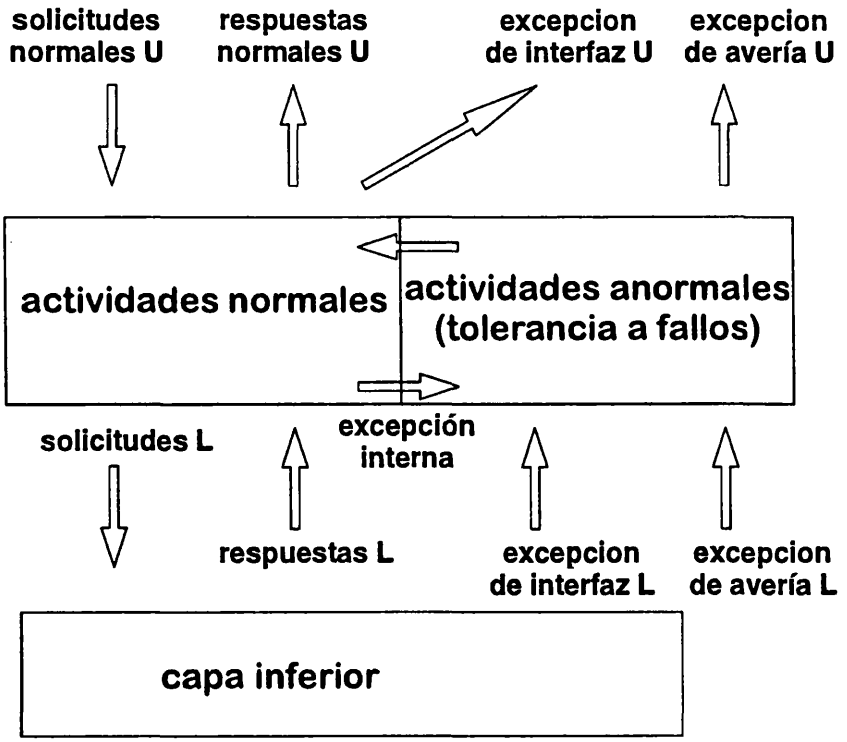


Fig. 5.1: *Componente idealizado tolerante a fallos*

5.1.1. FALLOS Y ERRORES

Recordando que el primer paso en el diseño de un componente fiable es la consideración del tipo de fallos que puede sufrir y cuáles debe tolerar, tipificaremos los fallos que pueden afectar a la comunicación serie, considerando de modo independiente cada uno de los sentidos de un canal de comunicación (aunque esta condición de independencia no se cumple estrictamente en la práctica). Los fallos que se pueden producir serán englobados en uno de los dos modelos siguientes:

- Fallo de **corte de línea**:

tipificamos como corte de línea aquellos fallos que supongan una interrupción permanente de la posibilidad de transmitir datos en un sentido, independientemente de la localización y características del problema. El fallo puede darse en el propio cable de comunicación (bien en la línea de envío o recepción, bien en la línea de tierra, bien en alguna otra línea utilizada por el protocolo de comunicación), en el puerto de comunicaciones del computador (con una variedad de posibilidades tal como sucede en el

cable de comunicación), en el conector que une el cable con el puerto del computador o en el equipamiento intermedio que pueda utilizarse, como por ejemplo modems.

• Fallo de **alteración de la transmisión**:

tipificamos como fallos de alteración de la transmisión aquellos que ocasionen cualquier tipo de modificación en los datos, fallos que principalmente estarán producidos por algún agente o perturbación externa en cualquiera de los puntos y niveles del subsistema de comunicación o por limitaciones propias del canal. La duración de estos fallos será la de su causa, pudiendo ser transitoria o permanente según las condiciones externas.

Hagamos notar que la alteración de los datos puede también producir que se modifique la estructura de la transmisión obteniendo aparentemente menos bytes de los esperados, pudiendo incluso interpretarse como una ausencia total de datos.

Notemos que esta clasificación de fallos no sería útil si no fuera acompañada de una clasificación de errores, dado que los fallos no son propiamente observables si no es por sus efectos (los errores o respuestas incorrectas del sistema). Es por ello necesaria la capacidad de determinar cuándo una respuesta es incorrecta y qué tipos de respuestas incorrectas pueden producirse (tipificación de errores) para poder diseñar la forma de detectarlos (métodos de detección de errores).

Los errores no tienen naturaleza propia, ya que siendo el error la manifestación de un fallo, fallos de distinto origen pueden tener manifestaciones similares, esto es, resultados iguales para el sistema de detección. Teniendo presente esta consideración, podemos definir dos tipos de errores para un sistema de comunicación:

(1) **Error de pérdida**:

se manifiesta como una reducción o eliminación de la información en destino (se reciben menos bytes que los enviados).

(2) **Error de corrupción**:

se manifiesta como una transmisión incorrecta de la información por modificación de uno o más de los bytes enviados.

Recordamos aquí que los canales de comunicación se clasifican según sus errores en [Swe91] canales sin memoria, canales simétricos, canales con ruido blanco y canales racheados, siendo estos errores tipificables en cualquier caso como errores de corrupción

Adviértase que es más importante esta tipificación de errores que la de fallos, y que ésta relación no es unívoca, pues mientras un fallo de corte produce siempre un error de pérdida, un fallo de alteración puede manifestarse como uno u otro tipo de error. Por su parte, visto desde el punto de vista de los errores, un error de corrupción viene siempre causado por un fallo de alteración, mientras un error de pérdida puede tener diversas causas.

La tabla de la figura 5.2 refleja la propuesta para clasificar por sus características cada uno de los tipos de fallos, así como el tipo de error producido y la latencia del fallo.

fallo	**corte de línea**	**alteración de datos**
tipo	hardware	*hardware*
causa	desgaste o accidente	perturbación exterior
duración	permanente	transitoria o permanente
error	(1) ausencia de datos	(1) ausencia aparente de datos o (2) datos incorrectos
latencia	hasta *time-out*	hasta *time-out* o hasta comprobación

Figura 5.2: *Modelo de fallos para un sentido o línea de comunicación*

5.1.2. ACTIVIDADES NORMALES

En este modelo que hemos elaborado, el componente "enlace de datos tolerante a fallos" recibirá solicitudes de servicios 'normales' relativos a la comunicación del mismo modo que las recibiría un sistema de comunicación sin propiedades especiales de tolerancia (básicamente éstas serán apertura, cierre, lectura, escritura y operaciones de control). Dentro de los servicios 'normales' se incluirán las solicitudes de servicio sobre el estado del

sistema de comunicación (dentro del grupo de operaciones de control). El componente atenderá estas solicitudes siempre que estas sean legales proporcionando el servicio solicitado y produciendo una respuesta (véase la parte izquierda de la figura 5.1).

El componente a su vez delega parte de sus funciones en uno o más subcomponentes que comprenden la capa física y el medio físico. De este modo, para las solicitudes que requieran la comunicación de datos, el componente solicitará servicios al subcomponente (capa física y medio físico), aislándose de este modo de los detalles del hardware.

Hemos considerado la utilización, tanto para las actividades normales como las excepcionales, del protocolo PPP (*Point-to-Point Protocol*) [RFC 1661] como base para la comunicación punto a punto, así como algunas de las opciones y extensiones previstas en este protocolo, como son los enlaces simples fiables [RFC 1663] y el protocolo punto a punto multienlace MLPPP [RFC 1771, I-D MP12], tal como se describen en el capítulo anterior. Sobre estos se incorporan variaciones para estudiar su influencia sobre la eficiencia de las actividades normales.

5.1.3. EXCEPCIONES

La interrupción de las actividades normales se produce por la ocurrencia de excepciones, que suponen el paso a las actividades anormales según se muestra en la figura 5.3.

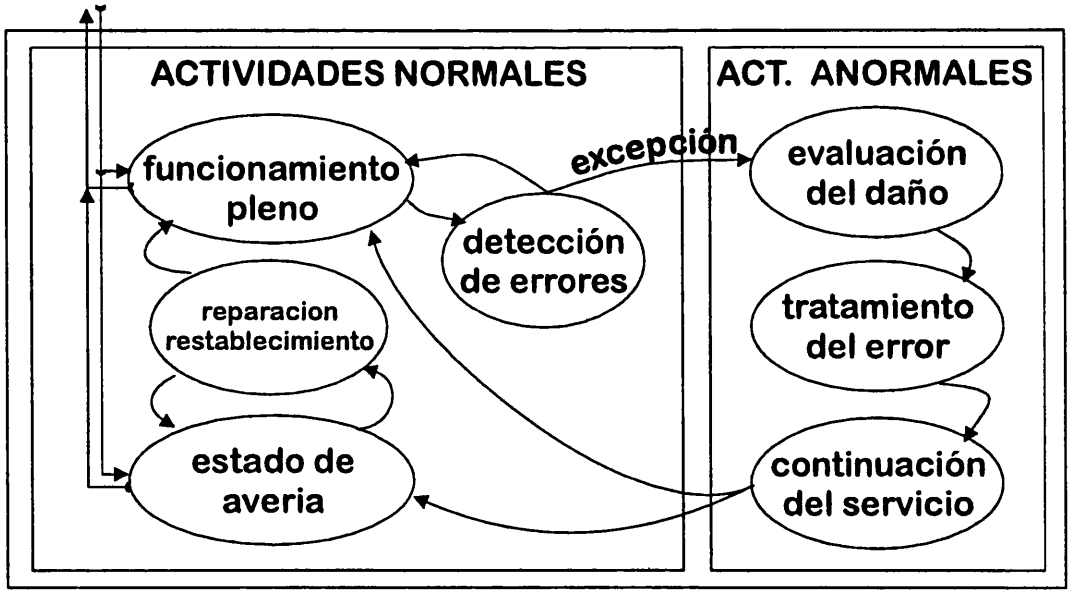


Fig. 5.3: *Esquema de actividades normales y actividades anormales*

Las actividades de tolerancia a fallos, que comprenden las excepciones y las actividades anormales, recaen completamente en el nivel del componente (nivel de enlace de datos), independientemente de que los subcomponentes utilizados incluyan o no características de tolerancia. Suponiendo que los subcomponentes a los que se confía parte del funcionamiento del sistema de comunicación (la capa física y el medio físico) no tuvieran ningún tipo de tolerancia (o no estén diseñados siguiendo esta metodología) estos será incapaces de detectar o corregir errores, presentándose sus respuestas como respuestas normales, o, en el peor caso, no produciéndose respuesta alguna.

Según el modelo de componente descrito en el capítulo primero y reflejado en la figura 5.1, las excepciones que conducen a la actividad anormal del componente se clasifican en tres tipos:

Excepciones internas. Las excepción interna se produce cuando el nivel de enlace de datos detecta un error de corrupción de los datos que no puede corregir directamente. Las actividades de detección de errores de corrupción de la transmisión son parte de la actividad normal del componente, comprobando los datos que constituyen la respuestas de los subcomponentes. La detección de un error de este tipo produce un cambio a la actividad anormal

Esta detección se producirá comprobando las respuestas normales de la capa inferior. En cualquier caso, esta comprobación también debe producirse incluso durante periodos en que la capa superior no solicite servicios, en cuyo caso será el propio componente el que solicite servicios a la capa inferior con el único objetivo de comprobar el buen funcionamiento de ésta por medio de sus respuestas.

Excepción de interfaz del nivel inferior. Esta excepción se provoca cuando se recibe una indicación de la capa inferior de que se ha producido algún tipo de error. Este tipo de respuesta del subcomponente es similar en naturaleza a la excepción de interfaz y, como aquella, debe provocar la actividad anormal del nivel principal.

Excepción de avería. La ausencia de respuesta en un plazo establecido se reconocerá como un error de pérdida que produce una excepción, produciéndose una señal ajena al componente (que provendrá del propio sistema operativo al expirar un temporizador, no siendo producida por el subcomponente como sucede en las excepciones de avería del componente idealizado).

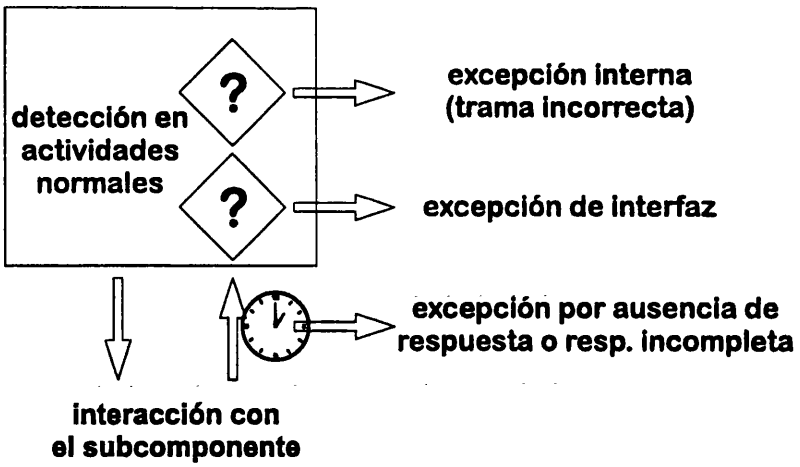


Fig. 5.4: *Excepciones y paso a actividad anormal*

5.1.4. ACTIVIDADES ANORMALES

Cualquiera que haya sido la manera en que se llegue a la actividad anormal, se deberá tratar de enmascarar el error, corregirlo, reparar y/o reconfigurar el módulo y volver a un estado de funcionamiento del componente según sus especificaciones (vuelta a la actividad normal). Si ello se consigue, las aplicaciones por encima del nivel de enlace de datos acabarán recibiendo respuestas normales, sin tener constancia de que haya producido error alguno, de manera que se cumplen los objetivos de tolerancia a fallos transparente.

Como se comentó en el capítulo tercero, algunos protocolos del nivel de enlace de datos como HDLC LAPB realizan actividades correspondientes a tolerancia a fallos, aunque la mayoría de las arquitecturas sitúan las actividades de tolerancia a los niveles superiores. En tales casos, la responsabilidad del nivel de enlace de cara a los niveles superiores se centra en evitar que los errores se propaguen hacia arriba. Para ello, los bloques de datos corrruptos son simplemente descartados, y si es necesario se reconfigura el sistema para seguir el funcionamiento de servicios normales.

Si el nivel no es capaz de seguir prestando servicios normalmente, según el modelo de componente idealizado, debe indicar al sistema continente (las capas superiores) la condición de avería, indicando que no le es posible resolver el problema internamente. A continuación volverá a su actividad normal pero en un estado de avería. Si tenía alguna solicitud en curso (habiéndose detectado el error durante el servicio a la misma),

responderá notificando un error del componente (equivalentemente, se puede considerar una excepción de interfaz hacia la capa superior).

En cualquier caso, cuando el sistema se haya averiado, contestará a todas las solicitudes de servicios indicando en con una respuesta de error la imposibilidad de atender las peticiones, excepto en el caso de que la solicitud sea una petición de información sobre el estado del componente. No obstante, el estado averiado puede no ser permanente. El componente podrá, de modo autónomo, efectuar operaciones de comprobación hasta llegar a decidir que la causa del error ha cesado (bien por cese de una perturbación externa, por sustitución o reparación de algún subcomponente, o por otras causas) y que es capaz de volver a ofrecer servicios, momento en el cual dejaría de contestar con códigos de error a las peticiones y las atendería debidamente.

5.2. DISEÑO DE SOLUCIONES TOLERANTES A FALLOS

En el sistema punto a punto que estamos analizando, las soluciones que adoptemos deben cubrir los siguientes objetivos:

- detección de errores,
- evaluación del error y del daño,
- confinamiento de los fallos,
- tratamiento del error,
- reconfiguración y continuación del servicio (vuelta a la actividad normal),
- reparación y recuperación

todo ello a un coste computacional (espacial y temporal) razonable.

5.2.1. MULTIPLICIDAD DE ENLACES

Como ya hemos citado en el apartado 4.3 el aprovechamiento de la capacidad de abrir múltiples conexiones entre dos pares, de manera que se combine más de un enlace PPP en un único enlace lógico, es el fundamento del protocolo MLPPP [RFC1717]. Los multienlaces, que surgen por la necesidad de poder aumentar el ancho de banda de un enlace a voluntad mediante la incorporación de nuevos canales, son también aprovechables para la tolerancia a fallos. En nuestras propuestas, consideraremos únicamente

procedimientos que utilicen la mínima redundancia de enlaces, dos, por lo cual hablaremos en ocasiones de enlace o canal duplicado.

5.2.2. MLPPP COMO CAPA DE RECUBRIMIENTO

En la arquitectura de enlace duplicado que se propone, las solicitudes de la capa superior son atendidas por el componente "servicio de enlace de datos" (MLPPP), que a su vez utiliza los servicios de los subcomponentes o capas inferiores (enlaces PPP simples), cuyos subcomponentes por su parte se consideran constituidos por los *driver* estándares de puerto de comunicación (nivel físico) y el medio físico. De este modo, las capas superiores interactúan en forma de solicitudes de servicio y respuestas con el servicio de enlace de datos sustentado por el protocolo MLPPP, el cual se comunica con las capas inferiores a través de los interfaces preexistentes.

Dada la existencia de una multiplicidad de canales, el hecho de solicitar servicios 'inferiores' a uno de ellos (o a varios simultáneamente) vendrá controlado por el protocolo MLPPP. Un esquema simplificado de la solución es el que se muestra en la figura 5.5.

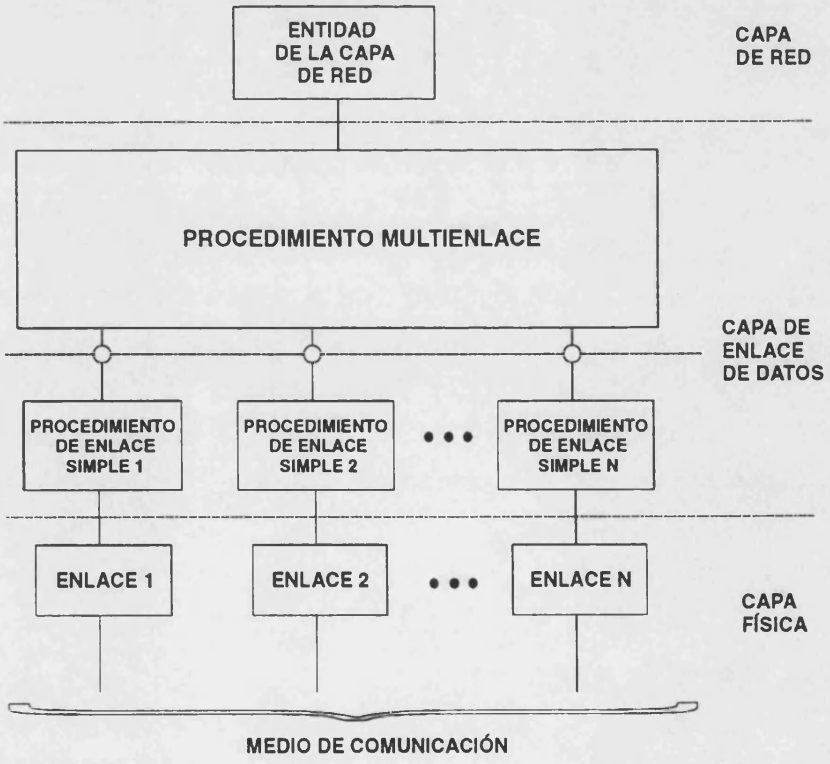


Fig. 5.5: *PPP Multienlace sobre enlaces PPP*

Un requisito del protocolo es que permita la transparencia en el intercambio de información (las aplicaciones que utilicen el servicio de comunicación tolerante no necesitan conocer los detalles del mismo) y a la vez la verificación del funcionamiento de los canales.

5.2.3. USO DE LA REDUNDANCIA TEMPORAL (ARQ)

Tanto si se usa un único enlace como si se usan múltiples enlaces, cada uno de los enlaces puede utilizar independientemente protocolos de enlace de datos con retransmisiones como LAPB al objeto de aumentar la fiabilidad propia para cada enlace. Es importante hacer notar que la utilización por los niveles superiores de cada enlace debe ser la misma independientemente de que el mismo utilice o no técnicas ARQ.

El uso de retransmisiones en el nivel de enlace de datos, además de la posibilidad de entrar en conflicto con retransmisiones de los niveles superiores (refiérase al apartado 3.3.6), permite tolerar fallos transitorios únicamente. Los fallos permanentes de un enlace punto a punto sólo podrán ser tolerados en este nivel por la existencia de otro (u otros) enlaces entre ambos nodos.

5.2.4. GESTIÓN DE LA REDUNDANCIA

La redundancia de recursos para la comunicación de datos puede ser aprovechada de diferentes maneras. La gestión de ésta redundancia, particularmente de la redundancia de canales propuesta en esta memoria, utilizando los modelos de tolerancia a fallos descritos en el capítulo primero, se expone en el siguiente apartado. El objetivo del análisis de la utilización de los distintos modelos en la comunicación punto a punto debe permitir discernir qué modelo es el más adecuado en función de las condiciones y requisitos de las aplicaciones que utilicen el servicio de enlace de datos.

5.3. EVALUACIÓN DEL DAÑO

La primera actividad que se realiza ante la detección de un error es la evaluación del daño producido. El alcance o extensión de un fallo, esto es, las zonas afectadas por el

mismo, se tratará de determinar según el tipo de error detectado, la forma en que haya sido detectado y la propia naturaleza del protocolo.

Consideraremos como **sentido de un canal** la entidad lógica constituida por

- las funciones software en el extremo emisor relacionadas con la salida de datos,
- el puerto de salida de este extremo emisor,
- el canal físico por el que se transmiten los datos,
- el puerto de entrada del extremo receptor, y
- las funciones en el extremo receptor relacionadas con la entrada de datos.

Esta entidad, mostrada en la figura 5.6, no es una entidad físicamente separable, pero será utilizada desde el punto de vista lógico para analizar el funcionamiento del sistema de comunicación. Cuando un extremo de la comunicación recibe una trama corrupta o el tiempo transcurrido sin recibir tramas supera un umbral (durante el cual el protocolo asegura la transmisión de una o más tramas) el nodo determina que se ha producido un fallo en el sentido entrante. En protocolos que utilizan asentimientos, cuando un extremo recibe una respuesta negativa a alguno de sus envíos o transcurre un tiempo límite sin recibir confirmación a un envío (pero en cambio sí que se reciben otras tramas) se determina que se ha producido un fallo en el sentido saliente.

Determinar con más exactitud la ubicación del fallo que ha provocado el error es una tarea que corresponde a un nivel inferior. Dado que nuestro objetivo es proponer la realización de todas las actividades anormales sin recurrir a hardware específico, se descarta el intento de ubicación concreta del fallo, considerando simplemente que un fallo producido en alguna parte de un sentido de un canal se extiende a todo el mismo.

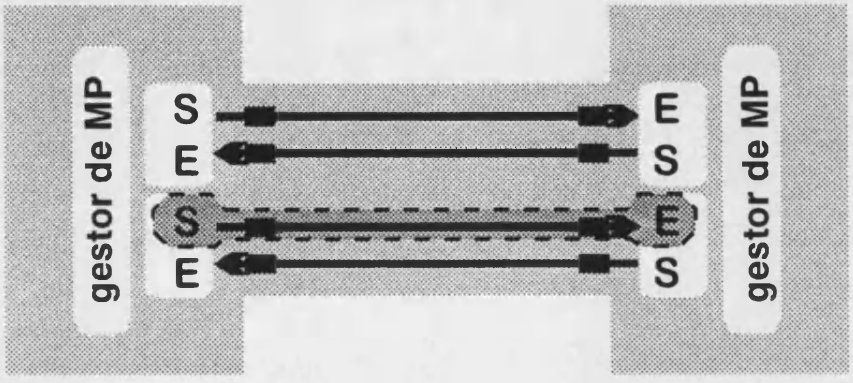


Figura 5.6: *Sentido de un Canal*

La evaluación del daño también comprende la determinación de la duración del mismo. En los protocolos basados en el tipo de fallos y errores del apartado 5.1.1. el tipo de error detectado no permite determinar con certeza el tipo del fallo para conocer si éste es transitorio o permanente. Por ello evaluaremos la duración de un fallo tanto por el número de veces consecutivas que se detecta un error como por el intervalo temporal desde que se detecta el primer error, estableciendo un umbral a partir del cual se considera que un fallo es de tipo permanente.

La utilidad de considerar la duración de un fallo es necesaria para determinar, según el protocolo empleado, si el subcomponente (sentido de un canal) en el cual se ha producido puede seguir siendo utilizado o si debe descartarse el uso del mismo mientras no se produzca reparación.

5.4. PROPUESTAS PARA EL TRATAMIENTO DE ERRORES

La actividad anormal posterior a la evaluación del daño es el tratamiento del error. Se considera como tratamiento del error el intento de corregir o enmascarar los efectos del mismo. En cualquier caso, la primera medida ante un error es el confinamiento del fallo que lo haya producido mediante el descarte de los datos que puedan haberse visto afectados por el mismo.

Para proporcionar tolerancia a fallos a la comunicación punto a punto con enlaces PPP aplicamos los modelos descritos en el capítulo primero, refiriéndonos a la ontología del apartado 1.5.3, con la siguiente correspondencia en los protocolos de enlace de datos:

Recursos: se refieren a los enlaces simples que unen los dos nodos adyacentes. También podrían considerarse como recursos diversos estilos de codificación o transmisión, aunque la consideración principal como recursos corresponde a un enlace simple o canal.

Acciones básicas: la acción elemental es la transmisión (envío/recepción) de una trama. Ello se debe a que los bits no son comprobados uno a uno, sino que, a efectos de poder gestionar y comprobar la transmisión, se agrupan en tramas que constituyen la unidad de detección y tratamiento de errores.

Resultado de una acción básica: el resultado de la transmisión de una trama debe ser bien un valor comprobable (normalmente se llama Secuencia de Verificación de Trama)

bien un valor comparable, aunque la comparación de tramas recibidas por medios distintos es una práctica poco usual.

Efecto de una acción básica: la transmisión de una trama tiene efecto en el extremo receptor en el momento de la entrega de la información a los niveles superiores y tiene efecto en el extremo emisor cuando la da por transmitida correctamente.

Punto de Recuperación y/o Partida: en el caso de tener que volver atrás o tener que avanzar, el punto de partida y/o recuperación será siempre el comienzo de una trama (esto es, en caso de pérdida o corrupción de la información, se retransmiten siempre tramas completas).

Temporizadores: se podrán establecer temporizadores de diversos tipos sobre la transmisión de tramas, aunque los más comunes serán los que sirvan para detectar canales inactivos (tiempo sin utilizar el canal), los que sirvan para detectar tramas incompletas (tiempo desde que empieza a recibirse una trama hasta que se reconoce el final de la misma) y los que sirvan para detectar tramas posiblemente perdidas en protocolos con asentimiento (tiempo desde que se envía una trama hasta que se recibe un asentimiento sobre la misma, que puede ser positivo o negativo)

Para apreciar la idoneidad de cada uno de los modelos propuestos valoraremos una serie de criterios, que incluyen las condiciones de transmisión y las condiciones resultantes de la aplicación de los algoritmos de tolerancia a fallos de cada modelo:

Como condiciones de la transmisión consideraremos:

- velocidad de transmisión
- frecuencia de los errores
- tamaño medio de las tramas (en contenido de información)

Como condiciones resultantes se consideran:

- tiempo medio requerido para transmitir una trama o
  número de tramas transmitidas por unidad de tiempo
- frecuencia de tramas perdidas (no entregadas a la capa superior)
- frecuencia de tramas entregadas con errores a la capa superior

Para cada modelo se hacen estimaciones de su efecto sobre las condiciones resultantes según las condiciones de trabajo.

Los dos primeros modelos (apartados 1.6.1.1. y 1.6.1.2.) buscan la tolerancia a fallos en la comunicación a través de cada enlace simple de comunicación mediante redundancia de información en el primer caso y mediante redundancia temporal en el segundo. Como ya se ha dicho, estos modelos no toleran fallos permanentes en los enlaces simples, limitándose a aumentar la fiabilidad de los mismos ante fallos transitorios. La utilización de estos modelos sobre los enlaces simples que forman parte del multienlace permitirá aumentar la eficacia de estos ante fallos transitorios.

Los modelos del 3 al 6 (apartados 1.6.2.1.1, 1.6.2.1.2, 1.6.2.2 y 1.6.3.1) utilizan multiplicidad de enlaces que son gestionados por la capa de recubrimiento MLPPP. El protocolo MLPPP recibe paquetes de los niveles superiores y los entrega a uno o ambos enlaces PPP según el modo de operación. En recepción, MLPPP acepta tramas llegadas a través de los enlaces PPP y entrega los paquetes pertinentes al nivel superior. (Ver figura 5.7)

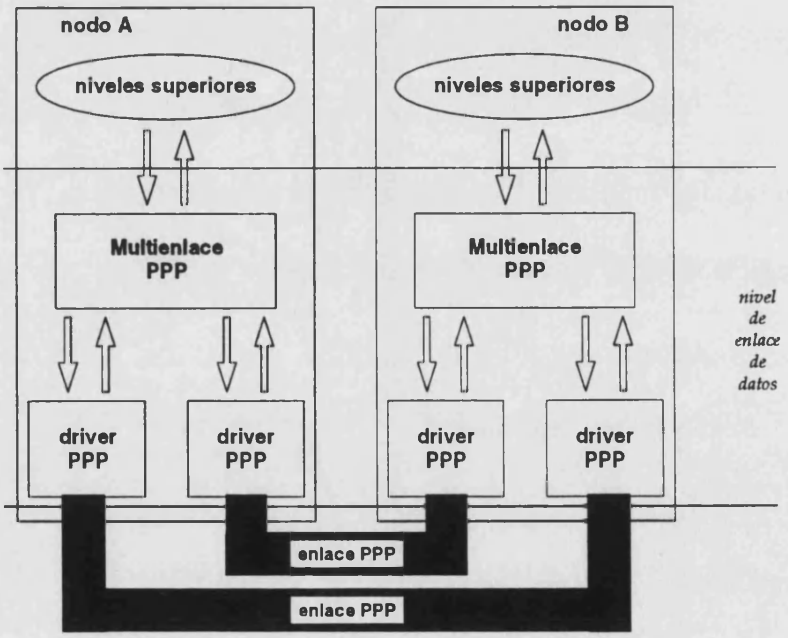


Fig. 5.7. *MLPPP para la gestión de recursos múltiples*

5.4.1. MODELO M1

El modelo M1 (apartado 1.6.1.1) se esquematiza como el uso de un solo recurso y corrección hacia adelante. Su aplicación supone la capacidad de corrección de los datos de una trama corrompida (lo que en la comunicación de datos se conoce como técnicas FEC transmitida a través de un enlace simple PPP. La práctica estándar de PPP consiste en añadir una FCS (*Frame Check Sequence*) únicamente para detección. La FCS en el protocolo PPP consta por defecto de 16 bits (2 bytes), pudiéndose negociar el uso de FCS de 32 bits (4 bytes), y dejando abierta la posibilidad de negociar otras longitudes de FCS.

El estudio de corrección de errores corresponde al campo de la teoría de codificación y debe considerar su coste en redundancia de la información [Rifa91]. En cualquier caso, precisa una redundancia de información mayor que los dos o cuatro bytes del FCS propuestos en PPP.

Puesto que el uso de técnicas FEC no produce un resultado libre de errores, incluso conociendo perfectamente las características de un canal y su frecuencia de errores, los sistemas de codificación empleados en este modelo M1 deben considerar en su diseño la frecuencia de errores resultantes aceptables. En caso de que se proporcione un bloque de datos corrupto a las capas superiores, éstas incorporan mecanismos adicionales de comprobación (normalmente sencillos *checksums* aritméticos) que detectan estos (ya sin ninguna capacidad correctora).

Este modelo trabaja con una sola transmisión por trama. Debe considerar el coste temporal de la corrección, aunque para velocidades de transmisión bajas es claramente inferior al tiempo de la transmisión. El ancho de banda requerido es pues el correspondiente a una trama, que se ve aumentado según se aumente el tamaño de la información redundante de la codificación.

Nótese que la corrección FEC sólo se puede intentar sobre la recepción completa de la trama, luego el error de pérdida no puede ser corregido. Por tanto la frecuencia de tramas no entregadas a la capa superiores vendrá dada por las tramas perdidas. Por su parte la frecuencia de tramas entregadas con errores vendrá condicionada por el tipo de codificación diseñada en lo referente a la distancia del código (número de errores a corregir y número de errores que debe simplemente detectar).

Este modelo permite incrementar la fiabilidad de cada enlace simple frente a errores de corrupción mediante el uso de mayor redundancia en la información. Sin embargo, como ya se ha comentado, estas técnicas requieren un coste computacional alto si deben

implementarse por software o la inclusión de hardware específico; es por ello que su empleo es relativamente poco frecuente comparado con otros modelos.

En resumen, el modelo M1 presenta las siguientes ventajas e inconvenientes más relevantes:

- *tolerancia a errores de corrupción sin necesidad de retransmisiones*
- *ninguna tolerancia a errores de pérdida*
- *mayor cantidad de información y alto coste computacional*

5.4.2. MODELO M2

Este modelo M2 (apartado 1.6.1.2) correspondiente a la recuperación por vuelta atrás sobre un único recurso, es ampliamente empleado en protocolos de comunicación que utilizan las técnicas ARQ (*Automatic Retransmission reQuest* o *Automatic ReQuest for replay*). Es directamente aplicable para la tolerancia a fallos en cada uno de los enlaces PPP por separado. Para ello, se negocia separadamente sobre cada enlace la opción de uso de tramas numeradas conformes al HDLC LAPB, que es la propuesta aportada para enlaces fiables PPP. Según las especificaciones esta opción de enlace fiable mediante el uso de HDLC LAPB debe ser acordada sobre el enlace antes de incluirlo en el grupo multienlace, lo cual limita la flexibilidad en la configuración del multienlace.

La recuperación por vuelta atrás sobre el enlace múltiple se trata también en los modelos posteriores que se refieren a la disponibilidad de múltiples recursos (los enlaces simples). La discusión anterior se contempla bajo este modelo por tratarse de tolerancia mediante el uso de un único recurso, el enlace simple fiable.

El ancho de banda requerido vendrá dado por el tamaño de la trama (información + redundancia; en este caso considerando sólo detección de errores) y el número medio de transmisiones correspondiente a cada trama.

El número de tramas no entregadas será menor que si no se utilizasen retransmisiones, pues sólo dejarán de ser entregadas las tramas que no hayan llegado correctamente al cabo de N intentos. El número de tramas entregadas con errores al nivel superior vendrá dado por la probabilidad de que una palabra del código sea corrompida hasta coincidir con otra palabra del código.

Las ventajas e inconvenientes principales de este modelo M2 son:

- *tolerancia a cualquier tipo de fallo transitorio*
- *el nivel de enlace de datos no pierde tramas a causa de fallos transitorios*
- *el uso de ARQ en este nivel puede entrar en conflicto con ARQ en niveles superiores*

5.4.3. HIBRIDACIÓN DE MODELOS M1 Y M2

Hay muchas maneras de combinar retransmisiones (ARQ) con técnicas de codificación para corrección de errores (FEC) produciendo modelos híbridos con características de ambos [Swe91]. La forma más obvia de disponer híbridos ARQ/FEC es usar un código para corrección parcial de errores, con detección de errores más severos, lo que se llama un híbrido de tipo I. El código sería un código corrector de errores normal pero usando únicamente parte de la mínima para corrección. Además, podría añadirse un código detector de errores adicional. Cuando los errores son detectados pero no corregidos, se necesita la retransmisión. La eficiencia será ligeramente menor que un ARQ puro para frecuencias de errores baja, por las cálculos adicionales requeridos, pero para frecuencias de errores altas, la eficiencia supera la de ARQ simple.

Los híbridos de tipo II utilizan códigos complementarios en las retransmisiones para mejorar la eficiencia respecto a los híbridos de tipo I ante frecuencia de errores baja utilizando sistemas de combinación de códigos o sistemas de combinación de la diversidad [Wic95]. Una posible solución de este tipo es que la primera transmisión se realice con bits de paridad para la detección de errores y en caso de que se precise una retransmisión esta consista en un código invertible con detección de errores. Un código invertible es aquel en el cual la información puede ser obtenida a partir de la paridad. Si la segunda transmisión se recibe correctamente, permite obtener la información, mientras que si se detectan errores en esta segunda transmisión, el contenido de la primera transmisión juntamente con la paridad de la segunda forman un código corrector de errores [Swe91].

Determinadas variaciones a la idea del híbrido de tipo II incluyen esquemas de códigos adaptativos en los cuales la razón (información_no_redundante / información_total) del código se incrementan cuando lo hace la frecuencia de aparición de errores.

Las ventajas e inconvenientes más relevantes de este modelo híbrido son:

- *la tolerancia a fallos de alteración se obtiene por codificación combinada con retransmisiones*

- *la tolerancia a pérdida de tramas se obtiene por retransmisiones*

- *el coste computacional de una implementación software es elevado*

5.4.4. MODELO M3

El modelo de repuestos durmientes M3 (apartado 1.6.2.1.1) supone la existencia de un enlace empleado para la transmisión de datos, sobre el cual se realiza detección de errores, y la existencia de uno o más enlaces inactivos que son utilizados para la transmisión cuando aparecen errores en el enlace primario. La utilización del enlace durmiente puede disponerse para ser utilizado tan pronto se detecte un error en el enlace primario, o bien después de una serie de errores detectados sobre el enlace primario.

El número de tramas no entregadas y el número de tramas entregadas con errores al nivel superior será el mismo que para un enlace PPP simple, diferenciándose de este por la posibilidad de reconfiguración y de utilización del enlace de repuesto.

Nótese que la existencia de un enlace de repuesto inactivo supone un claro desaprovechamiento del mismo, que sólo tiene justificación cuando el uso activo de un enlace implica un coste mayor que cuando éste no se utiliza (p.e. cuando se establece a través de una comunicación telefónica convencional). En caso de enlaces de repuesto instalados en una red privada, cuyo uso activo no supone coste adicional, el ancho de banda instalado es el doble del requerido para un único enlace. En este último caso, las condiciones resultantes son obviamente mejoradas con la aplicación de otros modelos tales como el M4 (réplicas activas) o el M6 (optimización de recursos múltiples).

El modelo M3 presenta las siguientes ventajas e inconvenientes:

- *comportamiento similar a un modelo de recurso único*

- *tolerancia a fallos simples permanentes por reconfiguración, requiriendo un tiempo para la misma*

- *desaprovechamiento de la red instalada cuando su uso no supone coste adicional*

5.4.5. MODELO M4

En el modelo de repuestos activos M4 (apartado 1.6.2.1.2.) los dos recursos realizan la misma función, por lo cual su aplicación a la transmisión de tramas supone que la misma trama es enviada por dos canales (caso de mínima redundancia, con un sólo repuesto activo). La información en recepción se toma del canal primario, y si ésta contiene errores, se toma del canal de repuesto activo. Este procedimiento es mejorado si en recepción se toma la información del primer canal que ofrezca los datos correctos, tal como se aprecia gráficamente en las figura 5.8 y 5.9. En este esquema resulta evidente que las tramas deben ir identificadas para que el sistema no asuma como información nueva los datos contenidos en una trama que ya se haya recibido por el otro canal. Esta es pues la exigencia de unicidad de efectos que se planteaba en el modelo.

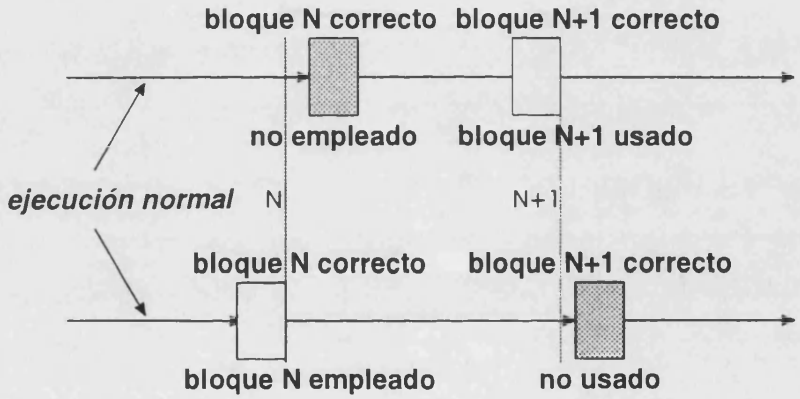


Fig. 5.8: *Selección de bloque en ausencia de fallos*

Aunque se use duplicación no se produce ninguna comparación entre las tramas recibidas por cada uno de los canales ya que ello supondría un retraso en la validación de cada trama recibida. La detección de errores se produce independientemente sobre cada trama. No existe un primario y un repuesto, ya que esta distribución de funciones no es aquí estática. No hay una asignación estricta de canal activo, sino que pasa por activo el primer canal en proporcionar los datos.

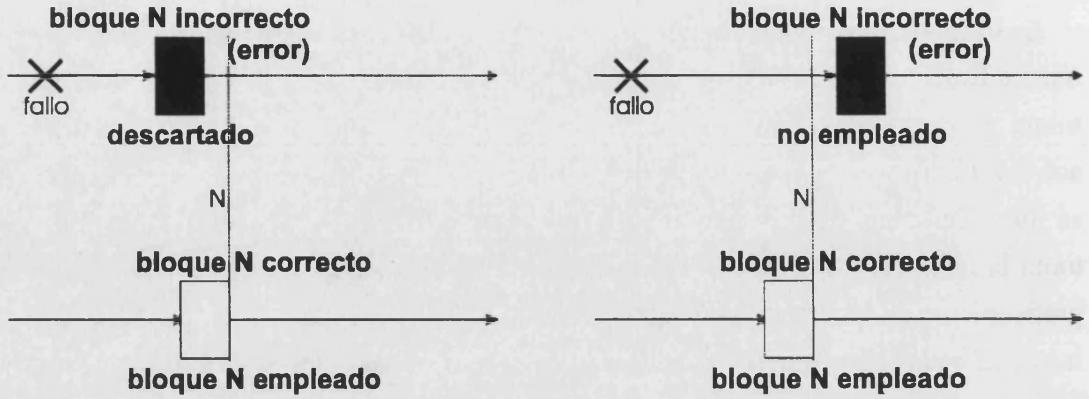


Fig. 5.9: *Selección de bloque en presencia de fallos*

La eficiencia del enlace en ausencia de fallos es la misma que la de un enlace simple. La diferencia reside en la frecuencia de tramas no entregadas a las capas superiores (tramas que hayan sido perdidas o corrompidas), no llegando una determinada trama correcta por ninguno de los dos canales, lo cual tiene una probabilidad igual al cuadrado de que suceda por un canal único. Por su parte la frecuencia de tramas entregadas con errores vendrá condicionada por el tipo de codificación y la probabilidad de que una palabra del código sea corrompida hasta coincidir con otra palabra del código.

Una hibridación de este modelo sería la negociación de fiabilidad sobre un enlace doble del tipo anterior, utilizando LAPB sobre un enlace que en realidad se duplica.

El algoritmo que hemos elaborado para implementar este modelo, para asegurar la unicidad de efectos en la comunicación duplicada sin retransmisiones en el nivel de enlace de datos, es el siguiente:

```
Procedimiento Transmitir Paquete
        duplicar el paquete
        marcar paquete como duplicado y numerarlo
        marcar paquete2 como duplicado y numerarlo igual
        entregar paquete a PPP0
        entregar paquete2 a PPP1
Fin Procedimiento

Procedimiento Recibir TramaMLPPP
        { cuando la trama MLPPP esta marcada como duplicado }
        identificar el numero de canal y el numero de trama
        si el numero de trama esta entre
            la ultima recibida por ese canal y la trama esperada
                descartar trama redundante ya recibida
                y salir del procedimiento
        extraer el paquete de la trama
        entregar el paquete a la trama superior
        esperar la trama siguiente a la recibida
Fin Procedimiento
```

La codificación de este algoritmo se incluye como modo TOLM_DUP dentro del módulo 'mlp.c' recogido en el anexo A y los resultados para la evaluación del mismo se encuentran en el apartado 7.4 y 7.5. Las principales ventajas e inconvenientes de este modelo M4 son:

- *enmascaramiento de fallos simples de cualquier tipo*
- *reconfiguración y restablecimiento automáticos*
- *rendimiento igual al de un enlace simple con menor tasa de errores*

5.4.6. MODELO M5

El modelo de réplicas con votadores M5 (apartado 1.6.2.2) implica un coste de recursos redundante mínimo de tres enlaces entre dos nodos adyacentes, y el ancho de banda requerido para la transmisión de una trama es triple que el de un enlace sencillo. Si la votación se realiza sobre cada uno de los bits recibidos, el porcentaje de errores con

votación entre tres es la probabilidad de que dos estén alterados más la de que tres hayan sido alterados $P(2) + P(3) = 3.(1-e).e^2 + e^3 = 3.e^2 - 2.e^3$ reduciendo pues la aparición de errores.

Sin embargo, un enlace de comunicación debe considerarse un recurso costoso, por lo cual la utilización de enlaces de comunicaciones a un tercio de su capacidad (en ausencia de errores, se utilizan tres enlaces para conseguir la eficiencia de uno solo) no está justificada por la exigencia de la fiabilidad excepto en sistemas especialmente críticos y, aun en éstos, es recomendable el uso de otras técnicas que requieran menos redundancia de enlaces y la utilicen eficientemente.

Las características más destacadas de este modelo M5 son:

- *enmascaramiento de fallos simples de cualquier tipo*
- *entrega de tramas erróneas en caso de fallos dobles iguales*
- *necesidad de excesivo número de recursos caros*

5.4.7. MODELO M6

El modelo M6 (apartado 1.6.3.1.) supone la utilización de todos los recursos disponibles para multiplicar la eficiencia del sistema. De hecho, la utilización de múltiples enlaces es la razón originaria de los multienlaces, donde se emplea ancho de banda bajo demanda (BOD) a coste adicional. En los procedimientos multienlace para aumentar el ancho de banda, la paralelización necesaria para aprovechar eficientemente la multiplicidad de recursos se obtiene mediante la fragmentación de los paquetes entregados por la capa superior. A los diferentes fragmentos se les adjunta una cabecera de secuenciación que permita su recomposición ordenada en recepción. Uno de los problemas fundamentales no resueltos por los estándares es cómo debe realizarse la distribución de los fragmentos a los enlaces. Los documentos de la IETF [RFC1717], se limitan a indicar la incertidumbre sobre el tiempo de entrega de cada fragmento, especialmente si se utilizan enlaces de distintos tipo, apuntando posibles estrategias como la de partir los paquetes en fragmentos proporcionales a la capacidad de cada enlace o partirlos en fragmentos iguales y asignar los fragmentos según la velocidad relativa de los enlaces.

Un problema específico en los multienlaces respecto al control de errores está asociado a la fragmentación, que supone que en el caso de pérdida o corrupción de un fragmento el protocolo MLPPP descarta todo el paquete. La posibilidad de que la perdida de un

fragmento se deba a un fallo permanente de un enlace obliga a establecer procedimientos específicos de control sobre los enlaces individuales, tales como la facilidad de monitorización de la calidad del enlace [RFC 1333] u otras acciones específicas como la petición periódica de ecos, para evitar la entrega de fragmentos a enlaces simples averiados. Consecuentemente, todo análisis del sistema debe considerar el número de fragmentos en que se divide un paquete.

Considerando la redundancia mínima (dos enlaces), según una estimación simplificada que contrastaremos en el apartado 7.5, en una conexión punto a punto con dos enlaces iguales, en ausencia de errores, se transmite una trama en poco más de la mitad del tiempo que lo haría a través de un único enlace. Dado que las FCSs se emplean para cada uno de los fragmentos por separado (no existen FCSs para el paquete reconstituido) la probabilidad de que datos corruptos lleguen a darse como correctos y entregarse a la capa superior es menor cuanto mayor sea la fragmentación, puesto que la corrupción tendría que producirse en todos los fragmentos alterados.

El algoritmo que hemos implementado para este modelo, considerando enlaces iguales, y sin retransmisiones en el nivel de enlace de datos, es el siguiente:

```
Procedimiento Transmitir Paquete
        crear un paquete2
        copiar la segunda mitad del paquete en paquete2
        truncar paquete por la mitad
        marcar paquete como principio de fragmento y numerar
        marcar paquete2 como fin de fragmento y numerar
        entregar paquete a PPP0
        entregar paquete2 a PPP1
Fin Procedimiento

Procedimiento Recibir Trama MLPPP
        { cuando la trama MLPPP esta marcada como fragmento }
        identificar el numero de canal y el numero de trama
        si el fragmento es la otra mitad del fragmento guardado
                si es un fragmento inicial
                        paquete = fragmento + fragmento guardado
                si_no
                        paquete = fragmento guardado + fragmento
                entregar el paquete a la trama superior
                esperar la trama siguiente a la recibida
        si_no si el numero de trama esta entre
            la ultima recibida por ese canal y la trama esperada
                descartar fragmento atrasado
                y salir del procedimiento
        si_no
                guardar el fragmento
                esperar la trama siguiente a la recibida
Fin Procedimiento
```

La codificación de este algoritmo se incluye como modo TOLM_DIV dentro del módulo mlp.c en el anexo A y los resultados para la evaluación del mismo se encuentran en el apartado 7.5. Las principales ventajas e inconvenientes de este modelo M6 son:

- *eficiencia cercana al doble de la de un enlace simple*
- *tolerancia nula a los fallos pero mayor confinamiento de los fallos*
- *posibilidad de transiciones a otros modelos*

A continuación vamos a describir dos variantes de este mismo modelo M6 que designaremos M6.I y M6.II que implementan la tolerancia utilizando retransmisiones en distintos subniveles.

5.4.7.1. M6.I: MULTIPLICIDAD SOBRE ENLACES FIABLES

Esta variación consiste en el uso dentro de un multienlace de enlaces fiables PPP [RFC1663] que utilizan retransmisiones en algunos enlaces simples por negociación independiente sobre cada uno de ellos. Ello permite aumentar la probabilidad de que los fragmentos entregados a los enlaces PPP fiables lleguen a su extremo, disminuyendo en consecuencia la probabilidad de tener que descartar paquetes completos.

El principal inconveniente de esta versión se da cuando se produce un fallo permanente en alguno de los enlaces o un fallo de mayor duración que el tiempo empleado para el número de reintentos establecido para el enlace fiable. Según los algoritmos específicados para PPP Multienlace referidos en el apartado 4.3.2. [RFC 1717 e I-D MP12], que establecen que los números de fragmentos asignados a cada enlace deben ser siempre crecientes, no está permitido reasignar los fragmentos asignados al enlace averiado, debiendo en consecuencia descartar todos los paquetes correspondientes a dichos fragmentos.

En esta versión, el tiempo necesario para transmitir un paquete es la mitad que en un enlace simple, multiplicado por el número medio de reintentos y corregida por el uso de cabeceras adicionales introducidas por la fragmentación. Respecto a la probabilidad de entregar paquetes corruptos, es la misma que en el caso anterior. Nótese que frente al caso general M6, esta versión incorpora tolerancia a fallos transitorios, pero a cambio aumenta los efectos de los fallos permanentes.

5.4.7.2. M6.II: FIABILIDAD SOBRE MULTIENLACES

La segunda variación consiste en el uso de retransmisiones en el subnivel de MLPPP sobre enlaces PPP no fiables. Esta constituye una solución que proporciona fiabilidad al multienlace en conjunto, de manera que el protocolo aplicado con los fragmentos, en caso de necesitar retransmisiones, permite su reasignación. El proceso de asignación tendrá en cuenta el estado de cada uno de los enlaces, intentando entregar el fragmento al enlace más fiable disponible en cada momento (que puede incluso ser el mismo), sopesando la fiabilidad con la ocupación del mismo teniendo en cuenta que el número de fragmentos que

deben asignarse a un enlace debe aumentar con la fiabilidad del mismo y disminuir con el grado de ocupación.

Esta aproximación supone la violación de la regla de PPP Multienlace según la cual los números de fragmentos asignados a cada enlace simple deben ser siempre crecientes. Sin embargo, el beneficio obtenido en cuanto a tolerancia a fallos transitorios de la conexión punto a punto justificaría esta transgresión, disminuyendo la probabilidad de descarte de paquetes por pérdida de fragmentos en enlaces inseguros.

Sin embargo, recordaremos que el uso de retransmisiones para corregir fallos transitorios se acostumbra a dejar para las capas superiores, y el uso de ARQ en el nivel de enlace de datos podría entrar en conflicto con las retransmisiones superiores, razón por la cual no implementaremos ninguna de las dos versiones anteriores.

En la tabla de la figura 5.13 se resumen las principales características que hemos analizado y que permiten una comparación entre los distintos modelos.

5.5. CONTINUACIÓN DEL SERVICIO

Tal como vimos en el capítulo expositivo de la metodología de tolerancia a fallos, la continuación del servicio incluye, antes de volver a cubrir los servicios normales, la posibilidad de reconfiguración del sistema, de reparación de elementos averiados y de restablecimiento del servicio de elementos reparados o recuperados. Analizaremos cada una de las actividades posteriores al tratamiento del error a la luz de los modelos que hemos planteado.

5.5.1. RECONFIGURACIÓN

Habiéndose detectado un error y evaluado el daño producido, en función de los resultados de dicha evaluación, debe decidirse cuál debe ser la configuración utilizada en las actividades normales del sistema de comunicación, indicando qué sentidos de cada canal de comunicación pueden ser utilizados. Por ello, tanto los módulos que rigen las actividades anormales como los que lo hacen con las actividades normales deben tener conocimiento de la disponibilidad o indicaciones de uso de cada uno de los sentidos de los canales de comunicación.

La reconfiguración debe pues determinar incluso si el sistema debe seguir empleando el mismo modo de funcionamiento para tolerancia a fallos o debe cambiar a otro modelo. Este cambio se realizará en función de las condiciones de trabajo, por lo cual la monitorización de los dos enlaces es necesaria para decidir el modo de funcionamiento a seguir. La figura 5.11. refleja como los datos obtenidos por medio de la monitarización de los enlaces simples conducen a seleccionar el modelo de gestión del multienlace según las condiciones de funcionamiento de los enlaces PPP.

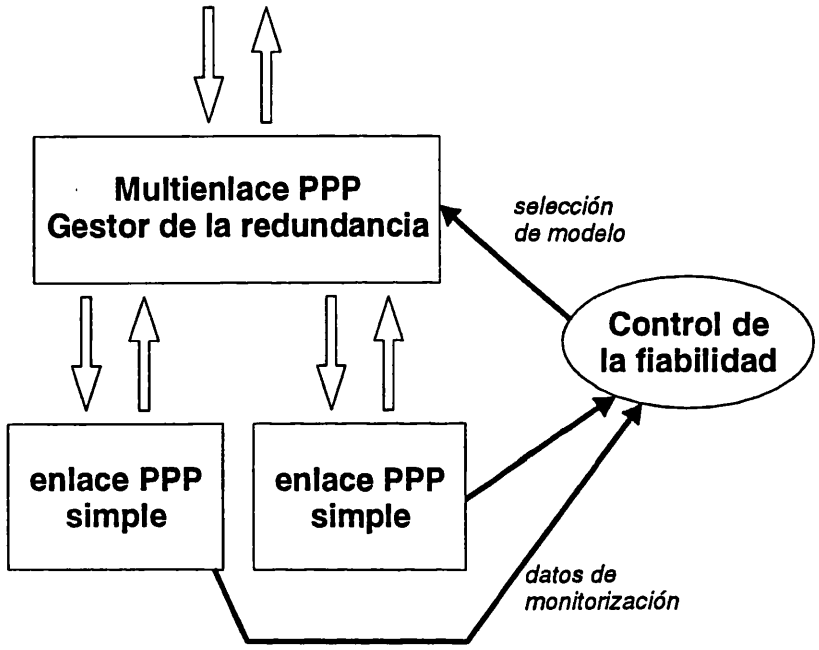


Fig. 5.11. *Monitorización de PPP para la selección del modelo de gestión de MLPPP*

Desde el punto de vista del proceso de reconfiguración, hay que destacar la propiedad de reconfiguración implícita presente en el modelo M4, derivada del hecho de que el sistema entrega la primera trama correcta disponible, con lo cual se asegura siempre la utilización de un enlace en funcionamiento. De los otros modelos, el modelo M3 utiliza reconfiguración por activación de un enlace de repuesto, requiriendo un cierto tiempo para esta activación. En los demás casos, la reconfiguración supone inevitablemente la transición a otro modelo.

Para profundizar en la selección de uno u otro modelo, consideremos el caso de un multienlace con dos enlaces simples gestionado según el modelo M6 que presenta la utilización más eficiente ante tasa de errores baja. La solución de reconfiguración más

apropiada para éste es la transición del modelo M6 al modelo M4 cuando se produce alguna de las siguientes circunstancias:

- se detecta un fallo permanente en un enlace simple,
- la tasa de errores de los enlaces simples supera un nivel umbral o
- el ancho de banda de un enlace simple es suficiente para el flujo de datos del sistema.

La reconfiguración también debe tener en cuenta si el sistema de comunicación es capaz de seguir ofreciendo todos sus servicios. Si se ha detectado un error irrecuperable sobre el único enlace habilitado, debe generarse una excepción de avería que se notifica al nivel superior (sistema continente). La versión de la figura 1.12, que mostraba las posibles transiciones para el caso de dos recursos disponibles, es aplicable a la gestión de enlaces de comunicación duplicados mediante los modelos M4 y M6 según la figura 5.12.

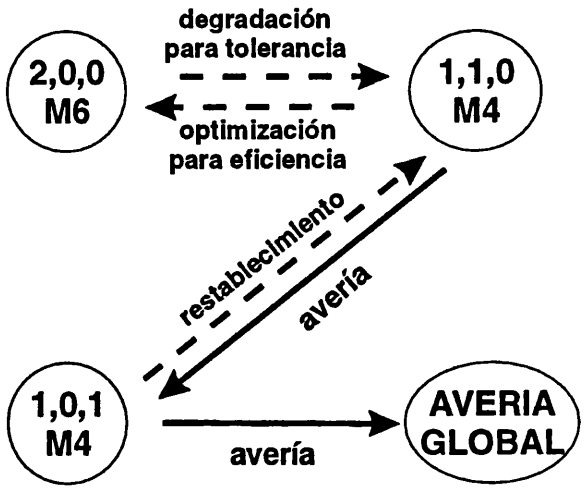


Fig. 5.12 *Posibles transiciones en MLPPP sobre dos enlaces PPP*

Es importante destacar que el modelo de gestión del multienlace puede ser diferente para uno u otro sentido de la comunicación. Por ejemplo, si un sentido se utiliza para la transmisión de un flujo importante de datos y el otro únicamente para respuestas cortas, el primero puede utilizar el modelo M6 ante errores transitorios poco frecuentes, mientras el sentido de las respuestas puede ser gestionado con el modelo M4.

El modelo de gestión de la transmisión a través del multienlace aplicado para un sentido de la comunicación se establece en el extremo emisor según los resultados de monitorización de los sentidos salientes. Por tanto, en un sistema que permite la transición entre modelos de gestión, el extremo receptor debe conocer el modelo de gestión de la transmisión o ser capaz de determinarlo a partir de las tramas recibidas, siendo esta segunda opción la que adoptamos en nuestra implementación incluyendo una marca a las tramas duplicadas.

5.5.2. REPARACIÓN

Cuando nos encontremos ante la presencia de errores repetidos, hay que plantearse un cambio cualitativo en el procedimiento de actuación haciéndose necesaria la reparación, consistente en la búsqueda y eliminación de alguna perturbación externa o en la sustitución física de un puerto de comunicaciones, de un cable o de cualquier elemento hardware sospechoso de producir el fallo. Ello supone la inclusión de un sistema que, a partir de la estadística de errores dispare una alarma que suponga el inicio de una actividad de campo ligada a los correspondientes equipos de mantenimiento.

La labor de estos equipos de mantenimiento será compatible con el funcionamiento del servicio cuando se use un modelo de gestión tolerante a fallos como es el modelo M4 y una arquitectura que permita la continuación de la comunicación a través de un enlace incluso durante las actividades físicas de mantenimiento y/o reparación. Una correcta política de comprobación periódica de los canales, permite determinar si la acción adoptada conlleva o no la reparación efectiva del canal.

5.5.3. RESTABLECIMIENTO

La inhabilitación de un sentido de un canal de comunicación fruto de la detección de un error en el mismo puede ser fruto de un fallo transitorio o de un fallo permanente, aunque esta circunstancia, como ya se ha expresado, es difícil de precisar con seguridad. Por ello es conveniente que los protocolos que inhabiliten sentidos de canales realicen actividades de comprobación consistentes en enviar tramas de control (sin contenido de información) a través de los canales, incluso de los sentidos inhabilitados. De esta manera, cuando desaparezca el fallo, el sentido erróneo empezará a responder correctamente y podrá volver a ser utilizado (proceso de recuperación).

Cuando el sistema ha pasado a un estado degradado, bien por causa de fallo o fallos transitorios, bien por fallos permanentes, la desaparición del fallo en el primer caso o la reparación en el segundo permite el restablecimiento del componente inhabilitado, recuperando para el servicio sentidos de canales anteriormente fuera de uso. El diseño debe decidir si dicha recuperación se produce automáticamente en el momento que desaparece o se repara el fallo, en otro momento determinado o fruto de una acción específica.

Del análisis de los modelos que hemos considerado, se deduce que sólo el modelo M4, reúne las características que permiten el restablecimiento automático. Para el resto de modelos, un módulo externo debe detectar cuando se ha producido la desaparición del fallo y restablecer el uso del enlace recuperado por transición a otro modelo.

5.6. CONCLUSIONES

Las principales características de los procedimientos de gestión de la redundancia en la comunicación punto a punto que hemos desarrollado se resumen en la tabla de la figura 5.13.

	PPP	M1	M2	M3	**M4**	M5	**M6**
número de enlaces necesarios	**1**	1	1	2	**2**	3	**2**
número de enlaces activos	**1**	1	1	1	**2**	3	**2**
eficiencia relativa	**1**	1--	1/(1-r)	1-	**1-**	1	**2**
corrección de 1 error de corrupción	**NO***	FEC	ARQ	NO*	**SI**	SI	**NO**
correción de 1 error de pérdida	**NO***	NO	ARQ	NO*	**SI**	SI	**NO**
tolerancia a 1 fallo permanente	**NO**	NO	NO	SI	**SI**	SI	**NO**
reconfiguración sobre el mismo modelo				SI	**SI**	NO	**NO**
reestablecimiento automático				NO	**SI**	SI	**NO**
tasa de perdidas	**r**	p	0	r	**r^2**	p^3	**r**
probabilidad de entregas incorrectas	**ε**	SI	ε	ε	**ε**	R^2	**ε^2**

Fig. 5.13: *Tabla resumen de características de los diferentes modelos*

La lectura de las primeras filas de esta tabla muestra como el modelo M6 es la opción preferente en ausencia de fallos en cuanto a la utilización de los recursos disponibles.

Aunque una lectura aislada de las filas sobre corrección de errores de la tabla de la figura 5.13 nos conduzca en principio a la conclusión de que las retransmisiones en el nivel de enlace de datos del modelo M2 proporcionan respuestas fiables, conviene recordar la discusión del apartado 3.3.6 donde planteábamos los posibles conflictos generados por el uso de retransmisiones en más de un nivel, para descartar el uso de ARQ en este nivel, permitiendo el uso de protocolos de nivel superior que incorporan ARQ por su naturaleza (tales como TCP) o por necesidades específicas de la aplicación (ARQ en el nivel de aplicación). El significado de las casillas NO* en la citada figura es que, aunque el error no sea corregido en el nivel de enlace, no hay inconveniente para que lo sea en los niveles superiores.

El establecimiento de un sistema de comunicación punto a punto tolerante a fallos, que mantenga la disponibilidad del servicio incluso en presencia de fallos permanentes de un enlace, exige la redundancia de enlaces punto a punto. Este capítulo ha propuesto distintos procedimientos para la gestión de la redundancia de recursos, que corresponden a las cuatro últimas columnas de la tabla. De ellos la tabla muestra como, en servicios de comunicación punto a punto donde se haga uso de forma permanente de dos enlaces simples, el modelo M4 es el que incluye mejores características de tolerancia a fallos.

En resumen, los procedimientos preferentes de gestión de la redundancia son los correspondientes a los modelos M4 y M6. El modelo M4, que prima la tolerancia, es más sencillo y en consecuencia más robusto, mientras el modelo M6, que prima la eficiencia, es más apropiado cuando el ancho de banda puede resultar insuficiente. La utilización de los modelos M4 y M6 en un caso de estudio se plantea en el capítulo siguiente y su implementación y resultados vienen recogidos en el capítulo séptimo y en el anexo A.

CAPÍTULO SEXTO

UN CASO DE ESTUDIO: SISTEMA DE CONTROL LINEAL DE LA VELOCIDAD

A partir de las aplicaciones ITS con requisitos específicos de fiabilidad en las comunicaciones descritas en el apartado 3.1.6. vamos a centrarnos en una aplicación concreta por la interacción existente con la misma y porque combina distintos elementos de interés. Se va a analizar en este capítulo el desarrollo de un servicio de ITS y la arquitectura del sistema necesaria para sustentarlo, centrándose en la fiabilidad del sistema y en los requisitos de comunicación, poniendo el énfasis en la utilización de enlaces fiables según la propuesta expresada en esta memoria de investigación, de modo que los enlaces fiables se integren en la red de comunicaciones global.

6.1. SERVICIO: CONTROL LINEAL DE LA VELOCIDAD

El estudio de este caso parte de nuestro conocimiento de diversos proyectos de control lineal de la velocidad en el ámbito europeo, especialmente un proyecto piloto desarrollado por la Agencia de Autopistas del Reino Unido. Implementaciones realizadas en Holanda y Alemania muestran como, por ejemplo, los alemanes han informado que la reducción del diferencial de velocidad entre vehículos, fruto del control lineal de la velocidad, ha supuesto una reducción del 29% de accidentes con heridos en la A-5 al oeste de Frankfurt. La Agencia de Autopistas del Reino Unido decidió implementar un proyecto piloto en el sector sudoeste de la M-25, la autopista orbital de Londres, para desarrollar la tecnología de control de velocidad y determinar las condiciones bajo las cuales se puede utilizar con mayor eficacia [Boy96]. Esta sección de la autopista consta de una calzada de cuatro carriles en ambos lados y representa una de las secciones de carretera más concurridas de Europa, con un tráfico de cerca de 200.000 vehículos diarios. El proyecto piloto fue presentado en 1995.

Este proyecto, aún en proceso de evaluación, muestra unos primeros datos alentadores sobre incremento de la fluidez y reducción de accidentes. Sin embargo, el sistema presenta, bajo nuestro punto de vista, dos defectos importantes:

- una estructura desarrollada específicamente para este sistema que no corresponde a una arquitectura global para los ITS, lo cual introduce algunos problemas ya comentados en los apartados 2.2 y 2.4;

- una diseño que, pese a incluir algunos mecanismos a prueba de fallos, está falto de una metodología global para dotar al sistema de control de tolerancia a fallos tanto en sus nodos como en sus comunicaciones, como se expresaba en el apartado 3.1.1.

En los siguientes apartados desarrollaremos una arquitectura para este servicio que supere estos dos defectos mediante la utilización conjunta de una arquitectura global para los ITS y de una metodología de tolerancia a fallos para sistemas de control.

6.1.1. DESCRIPCIÓN

El funcionamiento controlado de una autovía es la actuación sobre los límites de la velocidad en la misma para condicionar la velocidad de los vehículos en función del flujo de tráfico u ocupación, al objeto de reducir las situaciones de paro y arranque y aumentar la capacidad total de la vía. En las vías muy concurridas, durante las horas puntas, son habituales las situaciones de parada y arranque conocidas como interrupción del flujo o colapso del tráfico. El sistema de vía controlada pretende evitar esta situación estableciendo un flujo de tráfico continuo y fluido, de modo que todos los vehículos se mantengan en movimiento a una velocidad uniforme al objeto de que los conductores puedan completar su viaje en el mínimo tiempo y de la forma más confortable posible. Al reducirse el frenado y la aceleración se consigue también una mejora en el consumo de combustible y una reducción de las emisiones y del ruido. De este modo, se actúa en las tres líneas de objetivos de los ITS: la mejora de los tiempos de viaje y de la confortabilidad del mismo, el ahorro de energía consumida por las necesidades de movilidad y la minimización del impacto medioambiental relacionado con las actividades del transporte.

6.1.2. FUNDAMENTOS DEL SERVICIO

Uno de los factores principales que causan el colapso del tráfico es la reacción en cadena que provoca un vehículo en una corriente densa de tráfico cuando se ve obligado a

frenar. Esto puede estar causado porque otro vehículo ha cambiado de carril o porque un vehículo más rápido se acerca mucho a otro más lento. Al reducir la velocidad del tráfico limitando la velocidad máxima autorizada cuando el flujo es alto, se reduce el diferencial de velocidad entre vehículos, estableciendo una situación en la cual todos los carriles transcurren a una velocidad similar. Bajo estas condiciones, no se obtiene ventaja alguna cambiando de carril para intentar adelantar. En consecuencia los conductores son alentados a permanecer en su carril manteniendo un flujo uniforme, siguiendo al vehículo que le antecede y respetando la distancia de seguridad.

6.1.3. INTRODUCCIÓN DEL ASPECTO SANCIONADOR

La actividad sancionadora forma parte del bloque de actuaciones desarrolladas por la Administración por las cuales, a través de su influencia en el comportamiento de los conductores, pretenden prevenir los riesgos o, en el caso del control de la velocidad lineal, favorecer la agilidad de los desplazamientos. En situaciones de tráfico denso, que son las que motivarán el establecimiento de límites de velocidad inferiores a lo habitual, para que la política sancionadora sea efectiva se hace indispensable implantar un sistema automático que, respetando las garantías procesales, permita obtener unos resultados que retraigan a los infractores. Es igualmente importante indicar a los usuarios la presencia de un detector y de un sistema de denuncias: el objetivo de la actividad sancionadora es antes disuasorio que recaudatorio, y la disuasión también se consigue por ese medio. También es necesario el funcionamiento continuo del sistema sancionador, dado que si la amenaza no se hace efectiva hace perder credibilidad al mismo, reduciendo su efecto sobre el comportamiento de los conductores y reduciendo en consecuencia el rendimiento global del sistema.

6.1.4. ASPECTOS LEGALES Y ADMINISTRATIVOS

6.1.4.1. Sobre la señalización variable

En las modernas políticas de control de tráfico como la que constituye este caso de estudio, los paneles de mensajes variables (PMV) constituyen uno de los recursos elementales para la transmisión de mensajes dirigidos a los usuarios de la vía, siendo el medio de señalización variable más importante de cuantos existen en la actualidad. Los mensajes, aunque son mayoritariamente de carácter informativo, también pueden ser de precaución, en cuyo caso su influencia en el comportamiento del conductor es mayor (p.e. hielo en la carretera) y por último de obligación o mandato, cuyo incumplimiento puede

generar la actividad sancionadora de la Administración. Éstos últimos, para ser vinculantes requieren que el mensaje se emita en determinadas condiciones tanto de representación como de características (p.e. visibilidad).

Diversos artículos del Real Decreto 13/92 (Reglamento General de Circulación) legislan que las señales que aparezcan representadas a través de pictogramas (dibujos) en los PMVs deberán ajustarse a la forma, color, diseño, símbolos, significado y dimensiones a las que aparecen en el Catálogo Oficial de Señales de Circulación y Marcas Viales, previendo específicamente la posibilidad de que las señales luminosas (p.e. las de PMVs) inviertan los colores de forma que los símbolos aparezcan iluminados sobre fondo oscuro no luminoso [Aur96]. Estos preceptos son relevantes especialmente cuando afectan a las señales de cumplimiento imperativo y son desconocidos por muchos conductores que entienden que por aparecer en un PMV se trata de meras señales de recomendación.

En la legislación internacional, son de especial trascendencia los acuerdos de la Convención de Viena de 1968 y posteriores acuerdos complementarios, sobre la armonización de los símbolos de señalización vial. La propuesta de 1993 como complemento a dicha Convención, propuesta aun en fase de revisión, cita explícitamente los PMVs para establecer que toda señal que aparezca en un panel y sea conforme a los símbolos e inscripciones prescritos en la Convención tiene el mismo valor (incluido el imperativo) que si fuera señalización fija. Además, esta propuesta de 1993 recoge nuevos pictogramas que por su naturaleza son específicos de la señalización variable, como por ejemplo el pictograma de congestión, ya recogido en el Catálogo español.

6.1.4.2. Sistema Integrado de Denuncias

Los elementos de un sistema integrado de denuncias son un equipo de observancia forzosa capaz de detectar automáticamente la infracción y registrar la prueba de la misma, y un proceso informático que permita agilizar la gestión sancionadora (desde la identificación del infractor hasta la automatización de los trámites administrativos implicados). La condición indispensable en la implantación de estos sistemas es que no mermen las garantías instauradas en favor de los ciudadanos, garantías que varían fuertemente de unos países a otros.

La legislación española actual, en su Reglamento de Procedimiento Sancionador en materia de Tráfico, establece que las denuncias deben ser formuladas por los agentes de la autoridad encargados de la vigilancia del tráfico, y ser notificadas en el acto a los denunciados, excepto que no pudiera hacerse por razones reflejadas en la propia denuncia.

La problemática que en la práctica podría generar la no notificación en el acto al responsable de la infracción, por el hecho de que la Administración únicamente puede identificar el vehículo y dirigirse al titular del mismo, viene resuelta por la obligación del titular de comunicar el nombre del conductor, según recoge la Ley de Seguridad Vial (y cuya constitucionalidad, en el sentido de no violar el derecho a no declarar, ha sido ratificada recientemente) [Aur96].

En el sistema de control lineal de la velocidad que se analiza, teniendo en cuenta las condiciones en que se realiza de tráfico muy denso y con varios carriles de circulación por sentido, la posibilidad de notificar en el acto es prácticamente nula, contraproducente para la fluidez del tráfico e injusta por cuanto resultaría imposible detener a todos los conductores que cometen la infracción. Además, al contrario de otro tipo de infracciones que se fundamentan en la apreciación de los agentes, las infracciones de velocidad ya se detectan y miden por medios técnicos: los cinemómetros. Este medio técnico es considerado idóneo siempre que cumpla unos requisitos en su aprobación y verificaciones.

La automatización de los pasos posteriores a la detección automática enfrentan dudas legales aun por resolver. El artículo 3 del Reglamento Sancionador abre un camino, apoyado en los avances técnicos, para la agilización de los expedientes sancionadores, permitiendo el procedimiento de oficio por la autoridad competente cuando tenga noticias de hechos que puedan constituir infracciones. Ello permite que un sistema de observancia forzosa con suficiente credibilidad permita iniciar expedientes, pues la autoridad adquiere directamente conocimiento de los hechos sin intervención humana.

6.1.4.3. El vídeo en la gestión del tráfico

La utilización del vídeo y de las imágenes digitales adquiere más importancia en los ITS como fuente de suministro de datos, como medio técnico para la detección o para la confirmación de incidentes [Mar95] pues presenta evidentes ventajas sobre otros medios de sensorización. Sin embargo, en España no existe regulación relativa a la instalación de cámaras de vídeo en las vías públicas para la ordenación del tráfico. Teniendo en cuenta las posibilidades técnicas de almacenamiento y procesamiento informatizado de las imágenes, los aspectos a regular serían los siguientes:

- autoridad responsable de la instalación y autorización de sistemas de videovigilancia;
- no intromisión en la intimidad de las personas, particularmente en la instalación de sistemas de vídeo en zonas habitadas;

- información al público de la existencia del sistema de videovigilancia;
- aspectos relativos al archivo, registro y destrucción de las cintas o imágenes digitalizadas;
- admisión de las imágenes, especialmente las imágenes digitalizadas, como evidencia de una infracción.

Nuestra participación en proyectos a nivel europeo que afrontan los problemas legales relacionados con el uso del vídeo en la ordenación del tráfico nos ha permitido constatar la diversidad de legislaciones e intenciones[1], debida al diferente grado de implantación de los sistemas de videovigilancia, y la necesidad de una armonización para el desarrollo de esta subrama de los ITS.

6.2. FUNCIONES

El servicio de control lineal de la velocidad se puede implementar mediante la realización de cuatro funciones que se expresan a continuación:

6.2.1. Obtención de los parámetros del tráfico

El sistema utiliza los elementos adecuados para obtener mediciones directas como intensidad, velocidad o ocupación y para calcular, a partir de los datos medidos directamente, parámetros como tiempos de recorrido, distancia entre vehículos, velocidad media, etc.

---

[1]Una encuesta realizada a las autoridades de Francia, Holanda, Bélgica, Reino Unido, Alemania, España e Italia muestra la diversidad de legislaciones. En todos los países excepto Italia se considera deseable la notificación automática de las infracciones aunque en la actualidad todos excepto Holanda exigen una comprobación manual. Mientras en Holanda e Italia debe comunicarse la sanción al conductor, en Alemania, Reino Unido, Bélgica, Francia y España, el titular del vehículo es responsable de identificar al conductor cuando la notificación no tiene lugar en el acto. Italia, Alemania y Bélgica adjuntan la evidencia a la notificación de la denuncia; Reino Unido, España y Francia la entregan si así se solicita; Holanda sólo presenta la evidencia si es requerida en los juzgados.

Aunque ninguno de estos países tiene legislado el procedimiento sancionador por el uso de imágenes (*video enforcement*) todos ellos, especialmente Reino Unido y Francia, tienen interés en su uso dando prioridad a la seguridad y la eficiencia en la detección de infracciones de velocidad y de semáforos en rojo.

6.2.2. Determinación de los límites de velocidad

La investigación de los ingenieros de tráfico ha establecido niveles umbrales a partir de los cuales el flujo puede colapsar y los umbrales a los que se deberá activar los límites de velocidad decrecientes para demorar o evitar el colapso del tráfico. Se trabaja actualmente en algoritmos más complejos que consideren tanto el flujo como la velocidad de los vehículos para determinar qué límite debe establecerse.

6.2.3. Señalización de la limitación

Los límites determinados por los algoritmos de la función de determinación deben ser transmitidos a los conductores mediante la señalización adecuada, que indique la limitación impuesta y sus causas, con indicación expresa de su carácter de obligatoriedad y de la existencia de una actividad sancionadora asociada.

6.2.4. Actividad sancionadora

Dado que, como ya se ha apuntado, los beneficios del control de velocidad se obtendrán mediante el seguimiento real de los límites impuestos, debe incorporarse una función para sancionar, con las garantías adecuadas, a aquellos conductores que incumplan la señalización en vigor comprometiendo en consecuencia la efectividad del sistema.

6.3. SUBSISTEMAS FÍSICOS

El sistema que permita la realización de las funciones requeridas se implementa mediante el siguiente conjunto de sistemas físicos, con indicación en cada caso del tipo de nodo según la clasificación propuesta en [RKR96]

- ETD: Estación de Toma de Datos (nodo tipo D)
- PMV: Panel de Mensajes Variables (nodo tipo D)
- EOF: Equipo de Observancia Forzosa (nodo tipo D)
- Regulador (nodo tipo C)
- Sistema Central (nodo tipo B)
- Sistema Sancionador (nodo tipo A)

La estructura general del sistema se muestra en la figura siguiente:

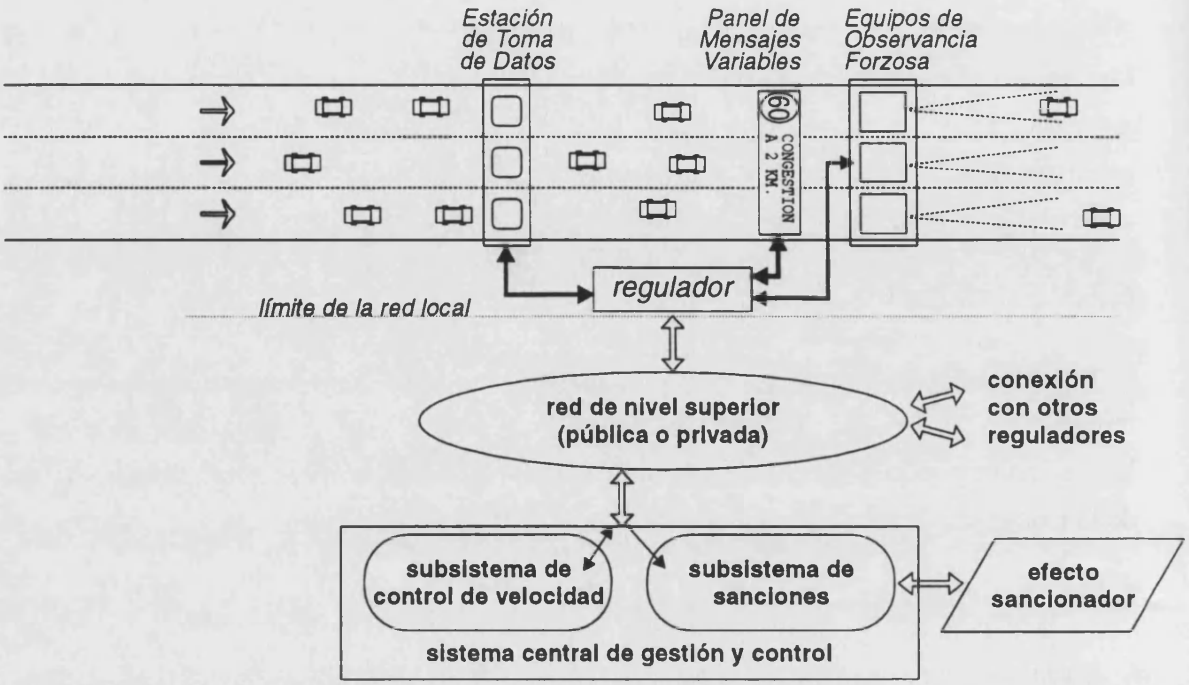


Fig. 6.1: *Sistema de control lineal de la velocidad*

Los diversos sensores, efectores, reguladores y sistema central deben incluir mecanismos a prueba de fallos o elementos redundantes para la tolerancia a fallos que garanticen su correcto funcionamiento. Dado que esta memoria de investigación se centra en la fiabilidad de las comunicaciones, no va a determinarse el diseño tolerante de cada uno de estos elementos, sino el efecto de una avería en cada uno de ellos o en sus comunicaciones.

6.3.1. REGULADOR

El regulador es el nodo fundamental en la gestión de un segmento de la red viaria. Sus funciones más relevantes son

- Gestionar la comunicación con los dispositivos de la red local: el regulador debe establecer y mantener la comunicación con los dispositivos locales, recibiendo sus datos o enviándoles órdenes, y monitorizar las condiciones de funcionamiento del enlace para detectar posibles fallos.

- Monitorizar los dispositivos de la red local: el regulador debe determinar si los dispositivos locales están funcionando correctamente. Para ello deberá controlar que los sensores envíen datos periódicamente y que los efectores respondan a solicitudes de estado. En caso de detectar alguna avería, debe comunicarla al sistema central.

- Cuando la red local contenga varios dispositivos de sensorización, el regulador debe integrar datos provenientes de distintas fuentes.

- Gestionar la comunicación con el sistema central: el regulador debe mantener y monitorizar la comunicación con el sistema central, y detectar el fallo en la comunicación con el mismo.

- Actuar como parte de un sistema de procesamiento distribuido: el regulador, en caso de que se produzca algún fallo en el ordenador central o en las comunicaciones con el mismo, el propio regulador de la red local debe actuar autónomamente estableciendo un límite de velocidad para el segmento que controla.

El regulador será un microordenador con un conjunto de puertos de comunicaciones dimensionado según el número de dispositivos en la red local. El sistema debe poder ser controlado local o remotamente. El regulador constituye un punto crítico del sistema, por lo cual debe tener especial provisión de tolerancia a fallos.

En caso de que se produzca una avería de un regulador, un segmento de la red viaria queda fuera de control, y el ordenador central deberá tener conocimiento del hecho para adoptar las medidas de mantenimiento oportunas.

6.3.2. ESTACIONES DE TOMA DE DATOS

El elemento base del sistema es el sistema de monitorización del flujo, que hace posible conocer los parámetros del tráfico medidos en campo. Los equipos ubicados en la carretera se denominan ETD (Estaciones de Toma de Datos) y las dos tecnologías más difundidas son las espiras electromagnéticas o con detectores de lazo inductivo y las Estación detectora por Visión Artificial o EVA

Las estaciones de toma de datos se sitúan a lo largo de la vía. Una ETD consta de uno o más elementos sensores conectados a un sistema con microprocesador situado a un lado de la propia vía, que registra los datos de sensorización directa: tiempo de paso (velocidad de

un vehículo), longitud del vehículo y ocupación del carril. Estos datos son procesados mediante algoritmos que permiten determinar los parámetros globales del tráfico, flujo y velocidad media de cada carril, y monitorizar las colas de cada carril, inclusive generar alertas cuando ocurren unas condiciones predefinidas.

La ubicación y cobertura de cada ETD dependerá del tipo de tecnología de sensorización. En el caso de ETD con lazos inductivos, en una autovía podrían situarse un par de lazos inductivos a intervalos de 500 metros sobre cada carril para la detección del paso de los vehículos. Para Estaciones de Visión Artificial, una ETD con una cámara podría instalarse cada 2 Km permitiendo controlar una sección de unos 500 metros (dependiendo del sistema de visión utilizado), obteniendo una cantidad de información mucho mayor que con los lazos inductivos [Mar95].

Las estaciones de toma de datos a partir del sistema con microprocesador, que es capaz de ejecutar un protocolo de comunicación simple como PPP, se conectan directamente a un regulador al cual transmiten periódicamente los parámetros del tráfico (sometidos a un mayor o menor grado de elaboración según la capacidad de proceso de la ETD y del regulador). Por motivos de fiabilidad, cada ETD deberá disponer de dos puertos de comunicaciones que se especificarán posteriormente.

La ETD debe monitorizar sus dispositivos sensores comunicando al sistema de procesamiento cuando se producen fallos en los mismos. Los sensores de las ETDs, por su ubicación de campo, son los elementos más propensos a los fallos. Sin embargo, la redundancia de los sensores resultaría excesivamente costosa, teniendo en cuenta el número de ellos y el hecho de que cada sensor dispone de otros sensores en posiciones cercanas al mismo. Los fallos se pueden manifestar por la ausencia de datos o por mediciones incorrectas. La tolerancia a fallos del sistema de sensorización es cubierta por el sistema de procesamiento que está preparado para trabajar con falta de datos o filtrando datos incoherentes.

Las ETDs deben poder ser programadas remotamente, bien para cambiar sus parámetros, bien para cargar un nuevo programa de funcionamiento.

6.3.3. SEÑALIZACIÓN (EFECTORES)

La orden de limitación de la velocidad se transmite desde el regulador, a través de la red local, a los Indicadores de Vía Controlada (IVC) cuya forma más simple son señales matriciales montadas en cada carril sobre pórticos especiales (pueden situarse a intervalos

de 1000 metros) y sobre postes en las vías de acceso a la vía controlada. A medida que el conductor se acerca a la zona controlada verá señales en todos los puntos de entrada indicándole que entra en una zona de velocidad controlada variable. El IVC. debe ser visto con facilidad (debe cumplir unas condiciones mínimas de brillo y ángulo de visión) y para que sea efectivo debe incluir, además de los dígitos que indican el límite de velocidad, el aro rojo alrededor que indica que se trata de una señal de obligatoriedad (según las normas internacionales sobre señalización variable).

La operatividad del sistema depende de la colaboración de los usuarios de la vía. Si el usuario no aprecia los beneficios que le reporta el sistema, tiende a incumplir, lo cual repercute negativamente en su funcionalidad. La experiencia demuestra que la tendencia a observar las normas es mayor cuando el conductor percibe que el mensaje que se emite es necesario o reporta alguna utilidad. Si un usuario entiende el sentido literal de la señal, pero no la razón por la que está situada en ese punto concreto y la considera inútil, tenderá a incumplirla. Por lo tanto es necesario informar al usuario del motivo y la utilidad de la señalización.

Por ello, el tipo de señalización variable más adecuado es el Panel de Mensajes Variables (PMV) debido a su alta operatividad, ya que pueden ser controlados a distancia o funcionar de manera autónoma y su versatilidad permite que junto a la propia señalización en forma de pictograma (el aro rojo rodeando la indicación numérica) aparezca un breve texto que indique su motivación.

Es necesario conocer con certeza la señalización vigente y realmente presente en cada momento. La avería de un efector debe ser conocida por el regulador para comunicarla a los operadores (y estos a los responsables de reparación y mantenimiento) y para que la detección de infracciones en una carril mal señalizado sea inhabilitada mientras no existan garantías del correcto funcionamiento de la señalización.

6.3.4. SISTEMA DE DETECCIÓN DE INFRACCIONES

El sistema incluye un módulo de observancia forzosa para la detección de infracciones como inicio de la actividad sancionadora.

Cuando un conductor que sobrepasa el límite fijado pasa por debajo de un pórtico con la señalización, salta el sistema de observancia forzosa y toma dos mediciones independientes, una mediante el radar que mide y registra la velocidad del vehículo y la otra mediante dos fotografías con un intervalo de medio segundo que permite calcular la

velocidad secundaria. La evidencia queda automáticamente registrada en la película con indicación de la matrícula (identificada por reconocimiento de formas sobre la imagen), número de la infracción, vía, ubicación del pórtico, sentido de la marcha, carril, fecha y hora, velocidad del vehículo según el radar, límite de la velocidad en ese momento y tiempo desde que se fijó el límite (el sistema incorpora un sistema de demora que no toma en cuenta el límite hasta un cierto lapso después de modificar la señalización para una reducción del límite, de tal modo que un vehículo que pase por el pórtico mientras cambia la señalización no sea sancionado, y que no sea necesario que los conductores frenen con brusquedad al observar un cambio del límite autorizado sino que lo hagan de manera gradual y segura).

Un segundo paso supone que, en lugar de grabar sobre película que debe ser recogida y procesada el registro de infracciones se haga mediante imágenes digitalizadas de una cámara que puedan ser transmitidas (preferiblemente comprimidas) a través de la red de comunicaciones hasta un ordenador central que gestione la actividad sancionadora.

El equipo de observancia forzosa debe ser fiable y duradero, por lo que se constituye con una importante provisión de tolerancia a fallos. El requisito de las dos mediciones independientes ya citados es de por sí una garantía de fiabilidad en la detección, ya que sólo en el caso de que ambas mediciones corroboren la infracción se aplica la sanción.

6.3.5. SISTEMA CENTRAL

El sistema central recibe los parámetros del tráfico de los diferentes segmentos de la red viaria gestionada por un regulador. El análisis del patrón del tráfico de la red viaria conduce a decidir el límite de velocidad adecuado para cada segmento. Los límites establecidos a nivel del ordenador central son comunicados a los reguladores correspondientes para que estos gestionen la señalización. El sistema central también participa en la actividad sancionadora recibiendo los datos de las infracciones cometidas y transfiriéndolas al sistema sancionador.

El sistema central debe monitorizar el estado de cada elemento del sistema, para tomar las medidas de mantenimiento y reparación que sean necesarias.

6.3.6. SISTEMA DE SANCIONES

Los datos de las infracciones detectadas y comprobadas son transmitidos al centro de gestión y control de tráfico, donde reside una base de datos que recoge las infracciones, desde el cual se establece relación con otros sistemas administrativos para la tramitación efectiva de la sanción.

Los datos de las infracciones son transmitidos cuando es posible hacia este centro gestor, sin restricciones de tiempo real, requiriendo una capacidad de almacenamiento que permita una transmisión periódicamente espaciada e incluso recuperación local en caso de que se produjese un fallo grave en la comunicación con el centro de control.

6.4. FLUJOS DE INFORMACIÓN

El conjunto de flujos de información relevante para el funcionamiento del sistema se refleja en la tabla de la figura 6.2:

Cod	Origen	Destino	Contenido	Periodo	Extensión	Tipo com.
F1	ETD	Regulador	parámetros del tráfico (+ estado de los sensores)	30 seg	reducida	local
F2	Regulador	PMV	límite de velocidad vigente y explicación	30 seg	reducida	local
F2r	PMV	Regulador	respuesta: estado del PMV	30 seg	reducida	local
F3	Regulador	EOF	límite de velocidad vigente	30 seg	reducida	local
F3r	EOF	Regulador	respuesta: estado del EOF	30 seg	reducida	local
F4	Regulador	S. Central	parámetros del tráfico (+ estado de la red local dispositivos y enlaces)	30 seg	media	red superior
F5	S. Central	Regulador	límite de velocidad y explicación	1 minuto	reducida	red superior
F631	EOF	S.Sancion.	datos de infractores	5 minutos	media	red local + red superior
F632	EOF	S.Sancion.	imagenes de infracciones	diaria *	extensa	red local + red superior
F11	ETD	Regulador	incidentes detectados	no periód	media	local
F411	Regulador	S. Central	incidentes detectados	no periód	media	red superior
F51	S. Central	ETD	control de parámetros, carga de nuevas versiones	no periód	extensa	red local + red superior
F51r	ETD	S. Central	respuesta	resp.	mínima	id.
F52	S. Central	PMV	carga de baterías de mensajes	no periód	extensa	red local + red superior
F52r	PMV	S. Central	respuesta	resp.	mínima	id.

Fig. 6.2: *Tabla de flujos de información en el sistema de control lineal de la velocidad*

La tabla de la figura 6.2. muestra como la mayoría de estos flujos de datos se producen entre dos nodos conectados directamente y en estos casos siempre se produce entre nodos de distinto nivel:

- entre la ETD y el regulador: F1 y F11
- entre el PMV y el regulador: F2 y F2r
- entre el EOF y el regulador: F3 y F3r
- entre el regulador y el sistema central: F4, F5 y F411

Sin embargo, algunos flujos deben realizarse entre nodos en principio no conectados directamente; en todos estos casos, los flujos se encaminan a través del regulador

- entre el EOF y el sistema sancionador: F631 y F632
- entre la ETD y el sistema central: F51 y F51r
- entre el PMV y el sistema central: F52 y F52r

Igualmente se puede apreciar diferencias en periodicidad y longitud:

- flujos cortos y más frecuentes en el tiempo correspondientes al funcionamiento continuo del sistema, que además se establecen entre nodos conectados directamente: F1, F2 y F2r, F3 y F3r, F4 y F5. Estos flujos deben ser gestionados por protocolos específicos de nivel de aplicación.

- flujos más extensos pero con periodos de ocurrencia más largos o no periódicos, que se establecen entre nodos no conectados directamente: F631, F632, F51 y F52. Estos flujos pueden gestionarse mediante protocolos de uso generalizado como FTP.

- flujos de ocurrencia no previsible pero que deben transmitirse en el menor tiempo posible: F11 y F411. Estos flujos también deben ser gestionados por protocolos específicos de nivel de aplicación.

Estos flujos se representan gráficamente en la figura 6.3.

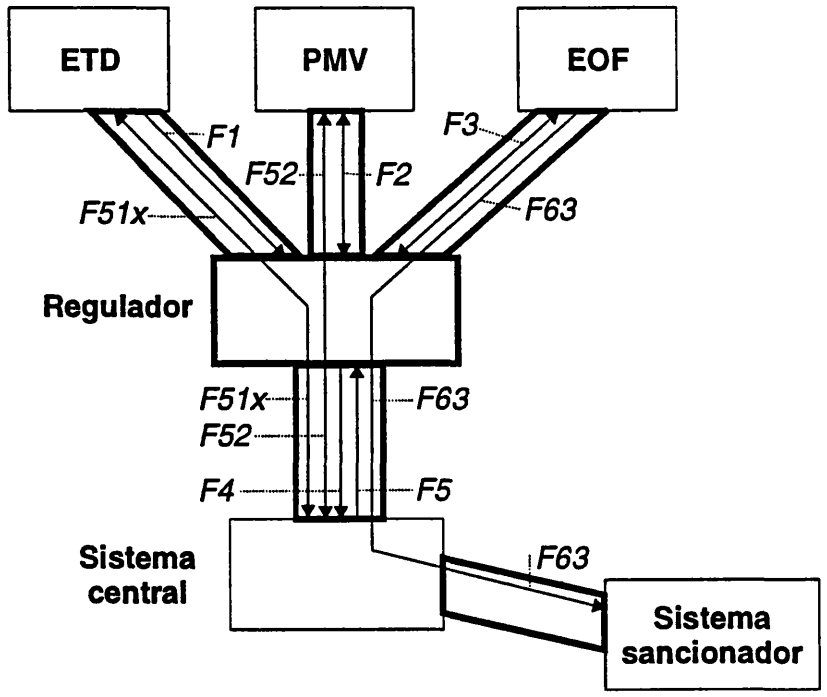


Fig. 6.3: *Flujos de datos en el sistema de control lineal de velocidad*

6.5. RED DE COMUNICACIONES Y FIABILIDAD

Los elementos necesarios para la transferencia de los flujos expresados en el apartado anterior son los siguientes:

Medio físico:

La comunicación en la red local entre el regulador y cada uno de los dispositivos (ETD, PMV, EOF) debe implementarse mediante enlaces punto a punto bidireccionales.

Dadas las características de la aplicación en cuanto a distancias entre nodos, los enlaces de la red local se implementan mediante conexión serie RS422. Dado el flujo de datos, se considera segura y suficiente una velocidad de 4800 bps. En caso de implementarse mediante puertos serie RS232, la distancia obligaría al uso de modems para cada puerto. Tanto en un caso como el otro se considera la utilización de par trenzado.

La comunicación entre el regulador y el sistema central se realiza a través de lo que hemos llamado red superior, y puede ser cualquier tipo de red de uso común, pública o privada.

Redundancia del enlace:

Los enlaces punto a punto de la red local estarán duplicados para aumentar la disponibilidad, y también debe estar duplicado el enlace entre el regulador y la red superior.

Los puertos serie redundantes deben estar aislados eléctricamente uno del otro para reducir la posibilidad de un fallo simultáneo de ambos por sobretensión que afecte a una tarjeta de comunicaciones o por cualquier otra causa externa.

Las líneas redundantes deben discurrir por canalizaciones diferentes si es económicamente viable, para disminuir la posibilidad de que una causa externa (p.e. unas obras) afecte a ambas al mismo tiempo.

Protocolo de enlace de datos:

Según se ha visto en los flujos de información, se debe utilizar un protocolo de enlace de datos balanceado, ya que aunque la mayoría de las comunicaciones serán iniciadas por el regulador o el sistema central, pueden haber incidentes que hagan preferible el inicio de la comunicación desde un dispositivo.

La existencia de versiones de PPP fácilmente disponibles para muchos medios, en particular para puertos serie, y la compatibilidad Internet que supone el uso de este estándar, aconseja el uso de este protocolo PPP, que permite que un servicio de enlace de datos proporcionado sea utilizable por sí mismo tanto para la comunicación directa en la red local (uso de los servicios del nivel de enlace de datos directamente por protocolos específicos de nivel de la aplicación) como para la comunicación con protocolos TCP/IP necesaria para la comunicación extremo a extremo fiable usando protocolos de aplicación generalizados, como FTP y Telnet, usados en la comunicación entre el sistema central a los dispositivos o entre el equipo de observancia forzosa y el sistema sancionador.

El uso del protocolo PPP permite además el uso del protocolo MLPPP como subcapa superior del nivel de enlace de datos utilizable para la gestión de la redundancia.

Gestión de redundancia:

La gestión de los enlaces redundantes mediante el protocolo MLPPP ofrece un servicio de enlace de datos con tolerancia a fallos de los enlaces simples y gestión dinámica que permite modificar el modelo de gestión según las condiciones y el tipo de transferencias necesarias.

Los modelos de gestión empleados serán M4 (apartado 5.4.5) y M6 (apartado 5.4.7) por ser los que presentan mejores prestaciones, tal como se describe en las conclusiones del capítulo quinto de esta memoria. La cantidad de datos en general no presenta problemas de ancho de banda, por lo cual la mayor parte del tiempo no será necesario emplear la redundancia para mejorar las prestaciones sino únicamente para aumentar la fiabilidad (modelo M4). Para aquellas transferencias menos frecuentes pero más extensas, como la transferencia de imágenes digitalizadas, que suponen una cantidad comprometida de datos, se favorecerá la eficiencia sobre la tolerancia (modelo M6) siempre que la monitorización indique que los dos enlaces están en funcionamiento correcto (en caso contrario, el multienlace debe permanecer en el modelo de gestión M4). Al finalizar estas transferencias, el sistema debe volver al modelo M4 que favorece la tolerancia. Recuérdese además que los modelos empleados para cada sentido de un enlace pueden ser diferentes.

El uso de un estándar Internet como MLPPP que soporta diversos protocolos de las capas superiores permite, como en el subapartado anterior, el uso de diferentes protocolos en los niveles superiores para la comunicación local y para la comunicación extremo a extremo.

La gestión de la redundancia por el modelo M4 permite mantener la disponibilidad del servicio aunque uno de los enlaces simples este averiado, y además posibilita las actividades de mantenimiento, reparación y restablecimiento descritas en los apartados 5.5.2 y 5.5.3 sin interrupción del servicio.

6.6. RESULTADOS

El incremento de la fiabilidad y disponibilidad de la comunicación punto a punto alcanzado con la utilización de multienlaces PPP con gestión variable en función de las condiciones y necesidades, contribuye al incremento de la fiabilidad y disponibilidad del servicio de control lineal de la velocidad dado que las situaciones de no disponibilidad de cualquiera de los enlaces entre regulador y dispositivos suponen una degradación del servicio bien por falta de datos de la situación del tráfico, bien por falta de acción sobre los

conductores, bien por ausencia de la actividad coercitiva, de modo que no se cumplen los objetivos de facilitar el tránsito. El uso de enlaces redundantes gestionados mediante MLPPP permite también las operaciones de mantenimiento y servicio continuado.

El segundo resultado destacable es que el uso de la interfaz proporcionada por MLPPP, favorece la interoperatividad al ser un estándar Internet y permite el uso de diferentes protocolos de las capas superiores, según las capacidades de los dispositivos para ejecutar protocolos de diferente complejidad. Por tanto los dispositivos de esta arquitectura son capaces de ejecutar, al menos, los protocolos PPP, MLPPP y un protocolo de aplicación (preferentemente un estándar como puede ser el STMF [NTCIP96s]). Los dispositivos con mayores capacidades de procesamiento deben poder ejecutar también protocolos TCP/IP y protocolos de nivel de aplicación Internet como Telnet y FTP.

6.7. CONSIDERACIONES SOBRE OTROS SISTEMAS ITS

Se exponen a continuación algunas consideraciones sobre la arquitectura y las comunicaciones que deben emplearse en otra de las aplicaciones ITS introducidas en el apartado 3.1.6, como ejemplo de la utilidad de los resultados del caso estudiado en otros sistemas de control del tráfico.

Un **sistema de carriles reversibles** precisa la fiabilidad de unas comunicaciones que permitan señalizar un carril como cerrado para un sentido para que al cabo de un tiempo prudencial se señalice como abierto para el otro (en una secuencia abierto-cerrado, cerrado-cerrado, cerrado-abierto). Es necesaria la certeza de que un sentido ha sido cerrado para iniciar una cuenta de espera antes de abrir el otro, aunque sin una especial criticidad temporal. Los enlaces de comunicación con la señalización podría utilizar enlaces dobles como en el caso de estudio tratado.

Las garantías de funcionamiento se centran en este caso en el aumento de la seguridad (*safety*) del sistema de carriles reversibles, para lo que se puede incorporar un sistema de detectores de vehículos. La misión de éstos es informar de la presencia y velocidad de vehículos en el carril para tomar decisiones sobre la señalización.

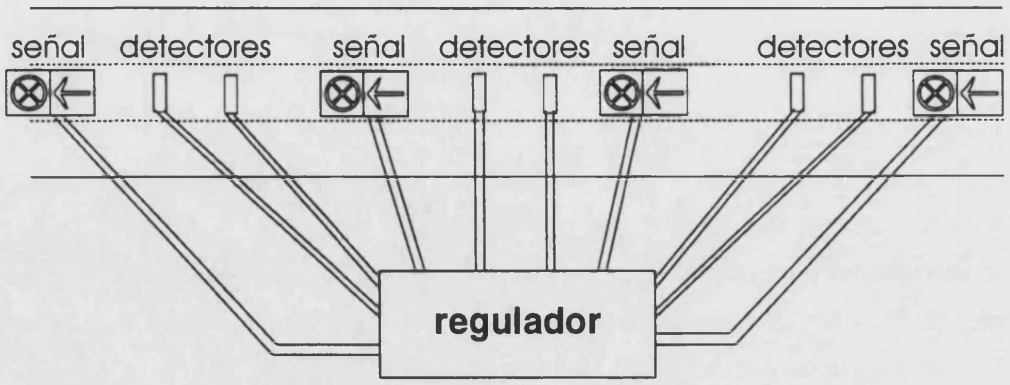


Fig. 6.4. *Sistema de carriles reversibles monitorizados*

La figura 6.4. muestra gráficamente los distintos elementos del sistema de carriles reversibles monitorizados, que brevemente se tratan a continuación.

SENSORES: el sistema incluye entre sus funciones la medición de los parámetros del tráfico. Los mismos elementos de sensorización empleados, sean simples lazos inductivos o complejas estaciones de visión artificial, son utilizados para la detección de vehículos circulando en sentido contrario al señalizado en un momento determinado.

EFECTORES: la actuación sobre el tráfico se materializa por medio de señales luminosas recogidas en la legislación, una flecha verde señalando verticalmente al carril indicando que se puede circular por él, un aspa roja indicando la prohibición de hacerlo y un flecha ámbar oblicua indicando que debe abandonarse el carril lo antes posible.

SISTEMA DE PROCESAMIENTO: el sistema de procesamiento tomará decisiones sobre el uso de un carril reversible según las condiciones del tráfico. El sistema de procesamiento, con o sin ayuda de automatismos especiales, debe garantizar la seguridad de la secuencia en un cambio de señalización. Adicionalmente, si el sistema de procesamiento recibe una alarma por detección de incidentes (vehículos en sentido contrario) debe señalizar inmediatamente el abandono del carril para todos los vehículos.

COMUNICACIONES: El correcto funcionamiento de la señalización solo puede ser conocido si la red de comunicaciones está disponible. Por ello, los enlaces entre regulador y señalización deben ser duplicados. Igualmente los enlaces entre detectores y regulador deben ser redundantes según las propuestas de esta memoria.

6.8. CONCLUSIONES

Este capítulo ha desarrollado un sistema de control de tráfico donde el sistema de procesamiento es tolerante a fallos. El sistema de procesamiento se considerará constituido por el conjunto de nodos y comunicaciones que, a partir de los datos adquiridos por los sensores, toma una serie de decisiones que materializa por medio de los efectores. La parte más sensible de este sistema de procesamiento es precisamente la red de comunicaciones, por lo que el uso de redundancia protectora de los enlaces de comunicación gestionada con los modelos que hemos desarrollado proporciona una fiabilidad al sistema mayor que si se utilizasen enlaces simples sin provisión especial de tolerancia a fallos.

Cuando los medios de comunicación tengan ancho de banda sobrante para las transmisiones necesarias, la gestión de los enlaces redundantes adoptada es preferentemente la más simple: los datos se envía replicados por los enlaces disponibles en cada caso (modelo M4 de gestión de la redundancia). De esta ingeniería que proponemos, que no va a ser implementada en esta investigación, desarrollamos en el próximo capítulo una implementación de MLPPP que permita comprobar los algoritmos propuestos de gestión de la redundancia.

CAPÍTULO SÉPTIMO

UNA IMPLEMENTACIÓN DE MLPPP

El presente capítulo describe la solución empleada para establecer un servicio de nivel de enlace de datos con tolerancia a fallos transparente para la comunicación punto a punto. La solución, válida para computadores con sistema operativo UNIX o similares, permite desarrollar y ensayar los algoritmos para comunicación tolerante a fallos expuestos en el capítulo quinto de esta memoria de investigación, mediante una plataforma para la implementación y comprobación de algoritmos, en un computador de propósito general sin necesidad de emplear ningún tipo de hardware específico.

Aun entendiendo que no todos los dispositivos de campo de una aplicación de ITS serán capaces de ejecutar sistemas operativos completos como UNIX, todos ellos al menos deben incluir los elementos básicos para la gestión de protocolos de comunicación que aquí describiremos específicamente para Linux, un derivado de UNIX de carácter público. Se va a analizar en primer lugar la organización del software que permite establecer una comunicación PPP simple sobre Linux [LAG][KHG] a través de un puerto serie, para después desarrollar la implementación del PPP Multienlace dentro del software del sistema y una serie de módulos adicionales para la gestión de la redundancia de enlaces.

Dado que partimos de que el hardware de E/S no proporciona la transparencia de uso que se persigue, es necesario implementarla por software con un subnivel emplazado sobre el servicio de nivel de enlace de datos mediante protocolo PPP disponible para casi todo tipo de computador. El subnivel se realizará modificando adecuadamente el PPP Multienlace para que incluya los algoritmos citados. Para que las capas superiores puedan utilizar el enlace de datos compuesto por un enlace múltiple como si fuese un único enlace lógico es por tanto necesario diseñar un módulo que gestione los enlaces apareciendo como un único dispositivo que ofrezca una interfaz estándar a dichos niveles superiores.

Además, dada la diversidad y especificidad del hardware de comunicaciones, viniendo cada tipo acompañado de su propio *driver*, se superpone la solución sobre los *drivers* existentes para PPP que ya tratan con el hardware, constituyéndose como una capa de recubrimiento sobre enlaces PPP que incorpora la tolerancia a fallos aislándose al mismo tiempo de los detalles particulares del hardware de comunicaciones.

7.1. REVISIÓN DEL SOFTWARE DE PPP BAJO LINUX

El interfaz de comunicación entre procesos (tanto si están en la misma máquina como si no) utilizado en los sistemas UNIX (Linux entre ellos) son los *sockets* de Berkeley [Rif92]. Este mecanismo proporciona transparencia total respecto a los protocolos utilizados en la comunicación. Un *socket* es un punto de comunicación por el cual un proceso puede enviar o recibir información. Para el proceso que maneja dicho *socket*, éste estará identificado mediante un descriptor de fichero, y la comunicación podrá realizarse utilizando las mismas funciones de lectura y escritura utilizadas con ficheros (write() y read()). Es más, el control de determinadas características de la comunicación se realizara a través de llamadas a ioctl() de la misma forma que se modificaban las características de acceso a fichero. Ademas de éstas, los *sockets* disponen de funciones adicionales mediante las cuales se pueden establecer: el tipo de protocolo a utilizar y características especificas del mismo, las direcciones de origen y destino, etc.

Las funciones del interfaz de *socket* forman parte del núcleo del sistema operativo (Linux en nuestro caso). La implementación de los *sockets* cubre los protocolos de red y transporte, estableciendo un interfaz con los protocolos de enlace de datos. En el caso de Linux, existen dos estructuras clave para la comprensión del interfaz entre la capa de red y la de enlace:

struct device: contiene las funciones del protocolo de enlace a utilizar, así como información del dispositivo físico que las va a llevar a cabo.

struct skbuff: es la estructura (colas) a través de la cual intercambiaran información ambas capas.

Cada protocolo de red seleccionará una estructura *device* para la transmisión de paquetes de acuerdo con su estrategia de encaminamiento.

Entre los diversos protocolos que se pueden seleccionar, nosotros vamos a utilizar los protocolos TCP/IP [Come94][Bla93] como protocolos de transporte y red, sobre el protocolo PPP como protocolo de enlace. [RFC1661][RFC1662]

Para el caso de usar IP como protocolo de red, la estrategia de encaminamiento de nuestras máquinas vendrá determinadas por las tablas de encaminamiento que podemos crear mediante el comando UNIX 'route'. De esta forma determinaremos que dispositivo usar dependiendo del destino. Normalmente la configuración de las tablas de encaminamiento se realizan durante el arranque de la máquina. En nuestro caso lo realiza el escrito de arranque rc.

El protocolo PPP tiene un status especial dentro del esquema que hemos visto hasta ahora. Por tratarse de un protocolo de enlace, tiene un interfaz a través de las estructuras *device* y *skbuff* similar al de Ethernet. Pero a diferencia de éste último, no está siempre disponible. Una conexión PPP necesita generalmente el establecimiento de una conexión entre modems y siempre realiza una negociación antes de permitir la utilización del enlace. Por este motivo, ademas de las rutinas existentes en el núcleo, la implementación del protocolo PPP usa un programa residente (*daemon*) encargado del establecimiento de la conexión y de la negociación. Solo si se llega al final de la negociación de red (IPCP en nuestro caso) con éxito estará el enlace disponible para la comunicación entre procesos que utilicen el protocolo TCP/IP.

Por otra parte, existe otra peculiaridad de la implementación de PPP. Así como los controladores de dispositivo de tarjetas de red no son utilizados como tales (a pesar de existir en /dev) sino a través de los *sockets*; PPP sí hace uso del controlador de dispositivo del puerto serie. Por ello tiene funciones para trabajar directamente sobre el *driver* escribiendo, leyendo o controlando su modo de funcionamiento.

A continuación se describen las estructuras y funciones disponibles en el núcleo de Linux, asociadas al protocolo PPP, reflejadas en la figura 7.1:

struct ppp: contiene información necesaria para la construcción de tramas PPP (carácter de escape, FCS, MRU, MTU, etc.), así como punteros a *buffers*, a la estructura de tipo *"struct device"* ya comentada y a otra estructura de tipo *"struct tty"* que permite el manejo del dispositivo de tipo carácter a través del cual se controla la linea serie (o el modem).

ppp_init(): es llamada al arrancar Linux y se encarga de inicializar una estructura de tipo *"struct ppp"* global que controlará el funcionamiento de este controlador de dispositivo.

ppp_open(): es llamada cuando se selecciona la disciplina PPP para un controlador de dispositivo. Vacía los *buffers* de lectura y escritura del controlador y realiza las inicializaciones necesarias para el nuevo modo de funcionamiento.

ppp_close(): libera el enlace PPP, retornando el controlador de dispositivo a la disciplina anterior (modo *tty*).

ppp_read(): lee datos de la cola de entrada del dispositivo

ppp_write(): escribe en la cola de salida del dispositivo.

ppp_ioctl(): controla el modo de operación del dispositivo.

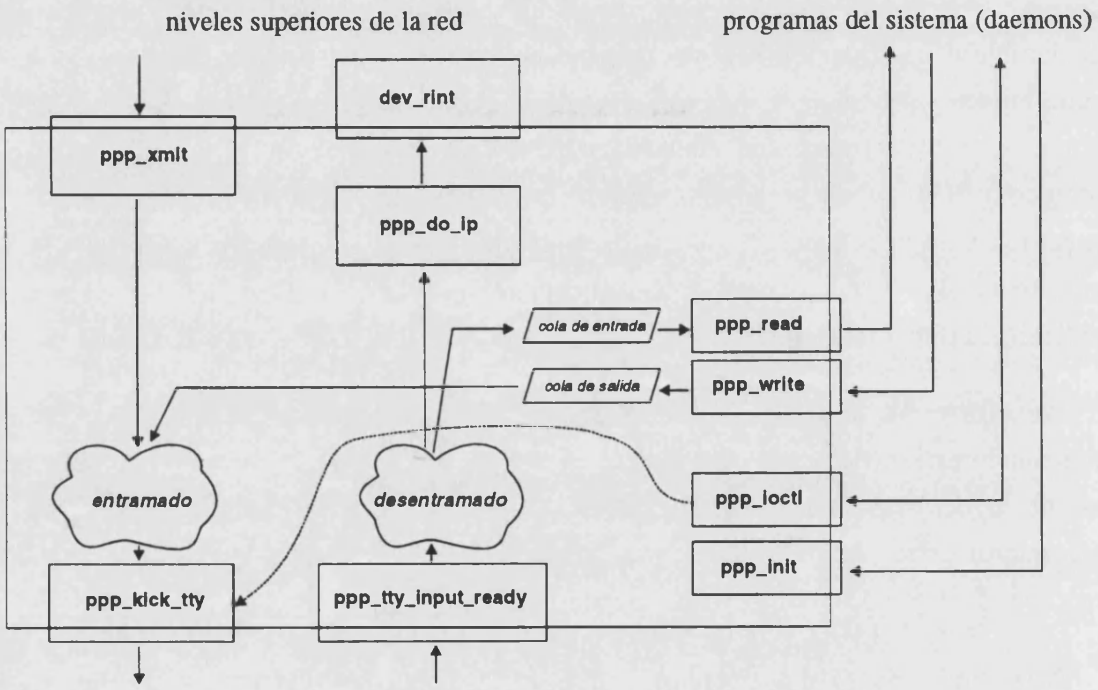


Fig 7.1: *Estructura del driver PPP v 2.0 en Linux*

ppp_xmit(): función del driver PPP que es llamada por los niveles superiores y recibe los paquetes procedentes de éstos. Realiza el entramado incluido el cálculo y adición de la SVT. y llama a otras funciones del driver para enviar el resultado por el puerto serie.

ppp_do_ip() : función del driver PPP que según el tipo de trama recibida pasa el paquete contenido en la trama a la función a *dev_rint()* que no es parte del *driver* PPP sino del nivel de red (es la función que entrega los paquetes correctos a los niveles superiores), o coloca la información recibida en la cola de entrada del dispositivo para que sea leída por los procesos del sistema.

ppp_kick_tty(): función del driver PPP que invoca la función *tty_write_data()* (ajena al driver PPP) para la transmisión de la trama formada a través del *tty* asociado a un puerto serie.

ppp_tty_input_ready(): función del driver PPP llamada por el nivel inferior cuando el tty tiene datos disponibles. Todos los datos que vienen por el puerto serie se reciben a través de esta función. A partir de esta se invocan otras funciones que extraen el paquete contenido en la trama, comprueban si la SVT. es correcto, y en caso afirmativo lo pasa a la capa de enlace a través del *skbuffer*.

A continuación se describe las acciones que lleva a cabo el *daemon* PPP, cuyo resultado (en caso de éxito) es un enlace PPP activo asociado a una ruta:

1. Establece las funciones encargadas de gestionar las señales que permitirán el comportamiento asíncrono del controlador de dispositivo.

2. Abre el dispositivo de tipo carácter asociado al puerto serie. En nuestro caso el dispositivo es "/dev/cua*n*".

3. Configura el controlador de dispositivo: selecciona baudios, paridad, control de flujo, etc.

4. Cambia la disciplina de linea del controlador de dispositivo a modo PPP. A partir de este momento, las funciones del controlador pasan a ser las propias de PPP. Es decir: la función de transmisión realizará el entramado adecuado incluyendo el SVT y pasando el resultado a la función de transmisión del controlador antiguo (modo *tty*), y la función de recepción realizara la operación inversa, guardando el resultado en el *skbuffer* adecuado para que la función de *socket* encargada de la recepción a nivel de red pueda llevar a cabo su misión.

5. Realiza la negociación de la capa de enlace de acuerdo con el protocolo LCP.

6. Realiza la negociación de la capa de red de acuerdo con el protocolo IPCP. Al final de esta fase, añade este dispositivo a las tablas de encaminamiento y se marca como activo el enlace. Cualquier paquete enviado a una dirección asociada a este enlace será pasado al controlador PPP (que forma parte del núcleo).

7. Espera a que un usuario local o remoto solicite la desconexión.

7.2. SERVICIOS OFRECIDOS POR LOS DISPOSITIVOS

El propósito de un *driver* de dispositivo en UNIX es atender las peticiones hechas por el núcleo del sistema operativo sobre un tipo particular de dispositivo. Un *driver* es un módulo de software que reside en el interior del núcleo del sistema operativo y que es la interfaz para uno o varios dispositivos. Este módulo está compuesto por una serie de funciones que son las que realizan las diferentes operaciones sobre el dispositivo.

Además de los dispositivos abstractos (entidades para el S.O.) que se corresponden directamente con dispositivos físicos, pueden definirse dispositivos que correspondan a servicios del sistema.

Esto permite diseñar cualquier dispositivo, por ejemplo un sistema de comunicación tolerante con multiplicidad de enlaces, que ofrezca un servicio de capa de enlace con comunicación fiable punto a punto con tolerancia a fallos. Este dispositivo se empleará en el resto de este trabajo para ejemplificar la plataforma propuesta.

Dotando a este dispositivo de un *driver* normalizado, la solicitud de cualquier servicio, con las mismas llamadas que formularía para comunicarse a través de un enlace PPP simple, y sin tener conocimiento explícito, accede a un servicio de enlace que le provee mayores prestaciones en las comunicaciones punto a punto. A su vez, estos servicios de capa de enlace están aislados completamente de la capa física, empleando para ello los servicios de enlace PPP establecidos para las comunicaciones simples, de manera que el desarrollo de un protocolo de Multienlace se independiza del conocimiento de los detalles específicos del hardware.

La estratificación del software por tanto consiste en interponer el control de Multienlace PPP entre el nivel de red y el control de enlace PPP con quien interactuaría en caso de tratarse de un enlace simple sin ningún procedimiento especial para tolerancia a fallos. Así pues, el control de Multienlace PPP constituye una subnivel superior dentro del nivel de enlace de datos, inmediatamente por debajo del nivel de red, interactuando por arriba con los protocolos de nivel de red y por debajo con los enlaces PPP [Con95], según refleja la figura 7.2.

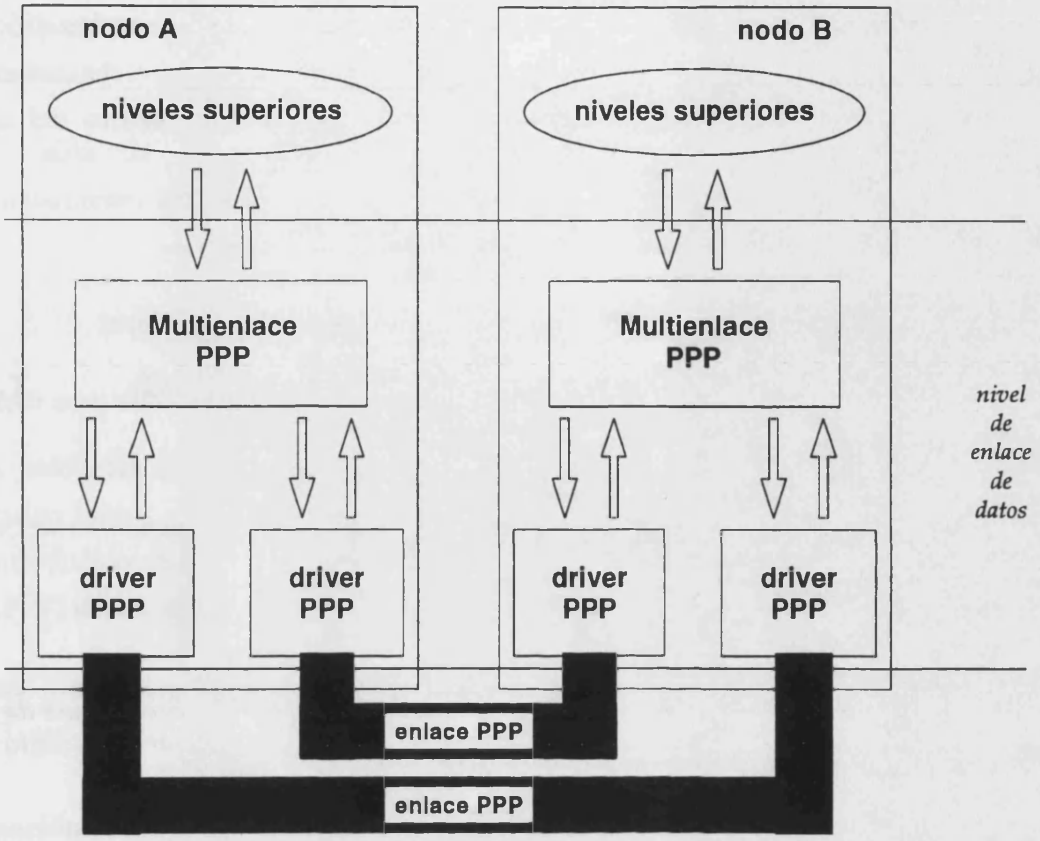


Fig. 7.2. *Estratificación del dispositivo del Multienlace PPP*

7.3. ESTRUCTURA SOFTWARE DE LA SOLUCIÓN EMPLEADA

Nuestro objetivo, como ya se ha dicho, es emplazar una capa de recubrimiento sobre un par de enlaces PPP, de modo que las capas superiores utilicen la interfaz de servicio de un nuevo dispositivo correspondiente a esta capa de recubrimiento.

La estructura del software desarrollada para la gestión de un multienlace PPP, estructura que queda reflejada en la figura 7.3, se aparta del concepto simple de un *driver* UNIX para estructurarse en dos tipos de módulos:

módulos residentes en el núcleo del sistema operativo

- *driver* PPP para un enlace simple, modificado ligeramente para aceptar las solicitudes de MLPPP.

- *driver* MLP situado sobre los *drivers* PPP existentes. Esta aproximación difiere de la práctica habitual de escribir un nuevo *driver* en el sentido que aprovecha los *drivers* existentes y coloca la tolerancia a fallos en un módulo específico dentro del núcleo [Eln93][Per97] que permite su adaptación en situaciones no estándar.

programas de nivel de usuario que se ejecutan fuera del núcleo

- programas que modifican la interfaz que utilizan los niveles superiores

- programas (*daemons*) de gestión simple de enlace PPP modificados para incluir el protocolo de monitorización LQP

- programas de monitorización y selección del modo de funcionamiento del *driver* MLP (y programas para la inyección de errores).

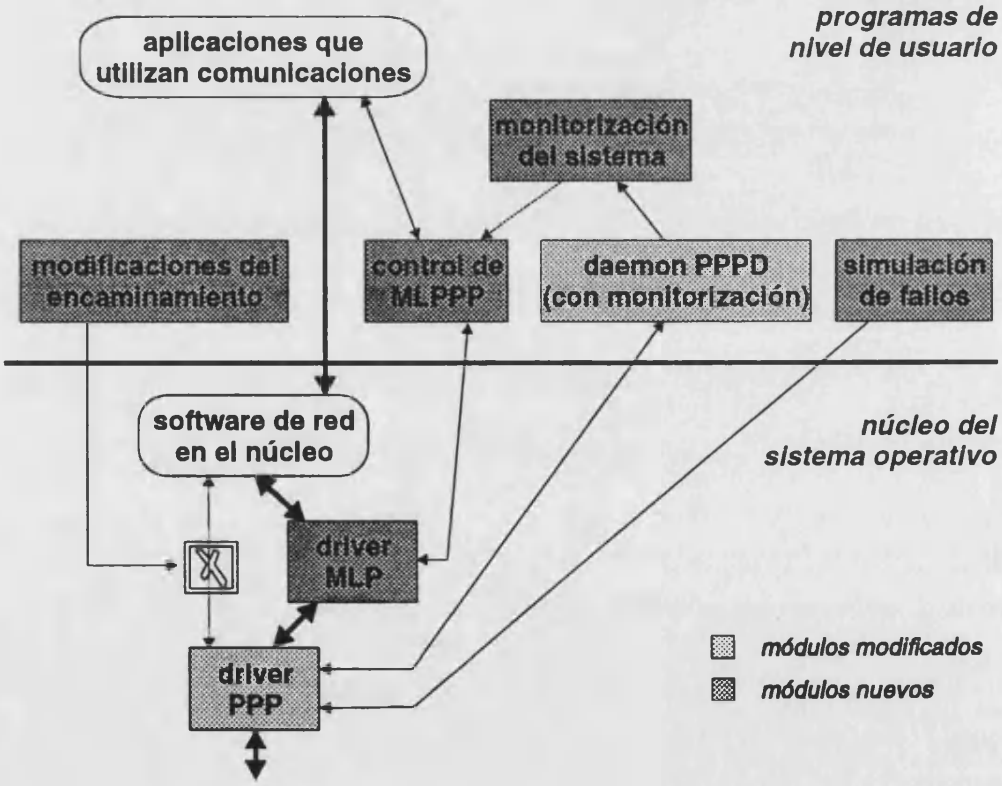


Fig. 7.3. *Estructura del software para la gestión de la redundancia MLPPP*

Esta serie de módulos constituyen una unidad en su conjunto análoga a un *driver* UNIX dado que cumple los requisitos funcionales de un *driver* [Cer96a]:

- proporciona funciones de servicios sobre los dispositivos a través de una interfaz normalizada;

- aísla de las particularidades del dispositivo, gestionando internamente el funcionamiento del dispositivo físico.

7.3.1. MODIFICACIÓN DEL DRIVER PPP

A partir del *driver* PPP existente (versión ppp-2.0) [RP20] en el núcleo del sistema operativo Linux, cuyo código fuente se contiene en el fichero $fuentes/drivers/net/ppp.c, se ha introducido una serie de modificaciones para permitir el uso de una subcapa superior (MLPPP) dentro del nivel de enlace de datos y por encima de dicho *driver*.

Las modificaciones introducidas respecto al código original, recogidas en el anexo A.1.3, son las siguientes:

Actuación sobre *ppp_do_ip*:

consiste en incluir PPP Multienlace entre los protocolos reconocidos de nivel superior. En caso de que una trama corresponda al protocolo MLPPP y el interfaz correspondiente al MLP esté activo, la trama es entregada al nivel del multienlace, correspondiente también al nivel de enlace de datos, en lugar de ser entregada directamente al nivel superior.

Actuación sobre *ppp_xmit*:

consiste en añadir la cabecera de MLPPP al paquete recibido de la capa superior. Estrictamente, esta acción debería realizarse dentro del *driver* MLP. Sin embargo, para reducir el número de copias del paquete (una primera copia en el *driver* MLP para añadirle la cabecera MLPPP, seguida por una copia para realizar el entramado PPP) el paquete se pasa intacto, indicando en la estructura de datos asociada (*struct skb*) que el paquete procede del protocolo MLPPP. De este modo, una misma función (*ppp_xmit*) inserta la cabecera MLPPP y realiza el entramado PPP (realizando una sola copia de los datos que constituyen el paquete).

Actuación sobre *ppp_kick_tty*:

aunque sobre esta función, que es la encargada de disponer la transmisión de los datos a través del puerto serie, no es necesaria modificación alguna, la actuación sobre ella ha sido

la de permitir alterar o impedir la transmisión física de los datos a efectos experimentales, al objeto de materializar la inyección de errores en el canal, actividad que se describe posteriormente.

7.3.2. DRIVER MLP

El *driver* construido para gestionar el multienlace proporciona a los niveles superiores las funciones de una interfaz normalizada de los dispositivos, produciendo la apariencia de estar trabajando con un dispositivo de comunicaciones simple.

Sus principales cometidos y funcionalidades son:

- servir las peticiones de los protocolos de los niveles superiores
- atender las peticiones de los programas del sistema que constituyen la parte complementaria del funcionamiento y
- gestionar el uso de los enlaces simples que forman el multienlace.

Nótese que el *driver* del multienlace, cuando trabaja según el modelo M4, actúa como capa de recubrimiento, ocultando los errores simples a los niveles superiores. Consecuentemente, el único caso en el que el nivel de red tiene noticias de los fallos que suceden en el nivel de enlaces simples es el de fallo doble y simultáneo de ambos enlaces.

7.3.2.1. MODOS DE FUNCIONAMIENTO DEL MULTIENLACE

Los modos de funcionamiento[1] implementados para el multienlace son cuatro, suficientes para nuestro objetivo, sin perjuicio de que se puedan implementar otros modos:

- envío de todos los paquetes a través del primer enlace (ppp0, modo TOLM_P0)

[1]Dado que el campo "metric" de la estructura de datos (struct device) asociada a todos los dispositivos de red, no tiene utilización para el dispositivo MLP en la versión de LINUX con la que se ha trabajado, dicho campo será el utilizado para determinar el modo de funcionamiento del multienlace.

- envío de todos los paquetes a través del segundo enlace (ppp1, modo TOLM_P1)

- envío duplicado de los paquetes a través de ambos enlaces (modo TOLM_DUP), correspondiente al modelo de gestión M4 (apartado 5.4.5)

- envío de los paquetes fragmentados a través de ambos enlaces (modo TOLM_DIV), correspondiente al modelo de gestión M6 (apartado 5.4.7)

7.3.2.2. FUNCIONES DEL *DRIVER* MLP

El código fuente del *driver* que gestiona el multienlace PPP reside en el fichero $fuentes/drivers/net/mlp.c recogido en el anexo A.1.1 y sus funciones principales, cuyo papel es apreciable en la figura 7.4, son las siguientes:

Función *mlp_ioctl*:

la función mlp_ioctl permite controlar el funcionamiento del dispositivo, bien para obtener datos de su estado y modo de funcionamiento actual, bien para modificar su modo de funcionamiento.

Función *mlp_xmit*:

la función *mlp_xmit*, cuando el interfaz del multienlace está activo, recibe los paquetes del nivel superior y, según el modo de funcionamiento del multienlace, los entrega a uno, otro o ambos enlaces PPP.

Tal como se ha indicado en la explicación de la función *ppp_xmit*, la función *mlp_xmit* no añade al paquete la cabecera de MLPPP, sino que incluye en la estructura *sk_buff* empleada para la interfaz con PPP (concretamente en los campos *pkt_type* y *saddr*) los datos necesarios para formar dicha cabecera.

Función *mlp_recep*:

la función *mlp_recep* recibe los datos correspondientes a una trama recibida correctamente a través de uno de los enlace PPP y los entrega como paquete al nivel superior (invocando a la función *dev_rint*), excepto en el caso de que se trate de una trama redundante (envío duplicado desde el otro extremo) cuyos datos ya han sido entregados al nivel superior.

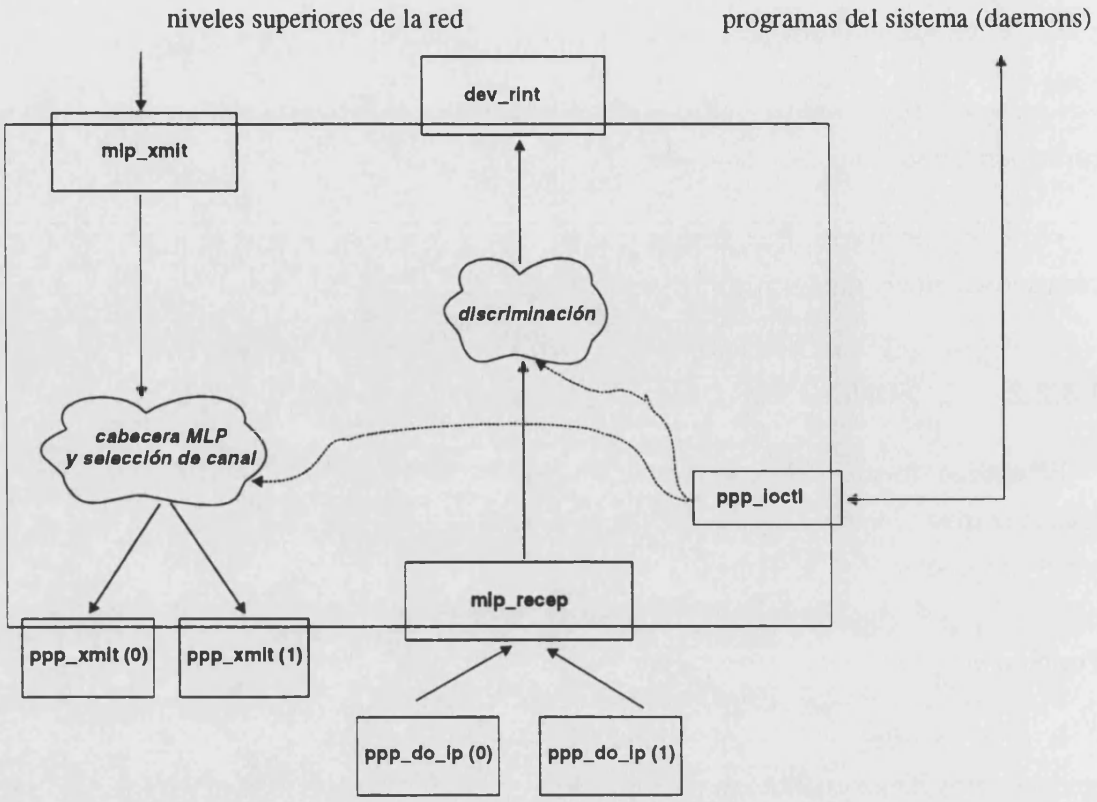


Fig. 7.4: Funciones del *driver* MLP

7.3.2.3. INCLUSIÓN DEL *DRIVER* MLP EN EL NÚCLEO DEL S.O.

La inclusión de un nuevo *driver* en el núcleo del S.O. Linux supone la modificación del fichero fuente $fuentes/drivers/net/Space.c para incluir el interfaz MLP en una lista que contiene todos las interfaces para dispositivos de red accesibles en el sistema y la modificación del *makefile* residente en el directorio $fuentes/drivers/net que contiene las fuentes de los *drivers* de dispositivos de red para incluir el módulo 'mlp' correspondiente al multienlace.

Nótese que esta inclusión supondrá que, tanto la primera vez que se incluya el *driver* MLP (y las modificaciones de PPP y de la lista de dispositivos) como cada vez que se modifique, la recompilación del núcleo incluirá la versión actualizada del *driver* MLP.

7.3.3. ENCAMINAMIENTO A TRAVÉS DE MLP

El sistema operativo que contiene en su núcleo el interfaz del MLP multienlace debe hacer que los niveles superiores utilicen dicho interfaz en lugar de la interfaz de enlace PPP simple. Para poder utilizarlo el interfaz MLP, debe estar ejecutándose un *daemon* pppd para cada uno de los enlaces PPP simples que formen el multienlace, y cada uno de estos debe establecerse previamente mediante el *daemon* pppd hasta alcanzar el estado 'abierto', descrito en el apartado 4.2.1, de modo que acepta paquetes de nivel superior.

A continuación se ejecuta un programa de nivel de usuario que hemos desarrollado, 'lanzamlp.c', cuyo código fuente se recoge en el anexo A.3.1. Este programa comprueba si existen enlaces PPP establecidos y se desactivan como interfaz de red los enlaces PPP abiertos. Si existía uno o más enlaces PPP establecidos se activa el interfaz MLP como interfaz de red único para la ruta entre los dos nodos extremos.

Es importante notar que este proceso sea reversible, mediante otro programa de nivel de usuario que hemos desarrollado, 'paramlp.c', cuyo código fuente se recoge en el anexo A.3.3 Su forma de trabajo consiste en desactivar el interfaz de red MLP para después activar las interfaces de red de los enlaces PPP simples por separado.

7.3.4. MONITORIZACIÓN DE LOS ENLACES

El protocolo punto a punto contempla la utilización de un protocolo de calidad para monitorizar de modo continuo el funcionamiento de un enlace PPP simple, tal como hemos comentado en el apartado 4.2.4.3. El intercambio de informes de la calidad del enlace (paquetes LQR: *Link Quality Reports*) por medio del protocolo estándar LQP (*Link Quality Protocol*) es una opción negociable de PPP descrita en [RFC1333].

La monitorización de un enlace supone un mecanismo (cómo monitorizar) y una política (qué hacer con los datos obtenidos en la monitorización). El estándar LQP [RFC1333] define explícitamente el mecanismo, y deja al implementador las decisiones subsiguientes. La estandarización del mecanismos permite la comunicación entre dos nodos aunque estos usen diferentes políticas respecto a la calidad del enlace.

La monitorización de los enlaces mediante el protocolo LQP requiere la modificación de las fuentes del *daemon* pppd que no incluyan este protocolo, debiendo modificar los ficheros 'main.c' y 'options.c', y crear un nuevo módulo 'lqp.c', cuyo código fuente se recoge también en el anexo A.2.2, que contenga las funciones *lqp_init*, *lqp_close*, *lqp_input* y

lqp_send que implementan el protocolo LQP y un fichero de cabecera 'lqp.h' (todos estos ficheros debe situarse en el directorio pppd que contiene los módulos fuente necesarios para generar el ejecutable del *daemon* pppd)

7.3.4.1. PROTOCOLO LQP

Para permitir diferentes políticas, LQP mide la calidad del enlace en pérdidas de bytes, de paquetes y de paquetes LQR. Cada medida se realiza por separado para cada sentido del enlace y es comunicada al otro extremo del mismo para que ambos extremos dispongan de información completa de la calidad del enlace.

Las implementaciones del protocolo LQP mantienen contadores de los items citados (bytes, paquetes y paquetes LQR) transmitidos y recibidos correctamente. Los contadores son monotónicos, y la calidad del enlace en un período se obtiene por diferencia entre los valores de los mismos. Los contadores utilizados son acordes con [MIB]. Cada uno de los contadores, el contenido de los paquetes LQR y la confección de los paquetes se refleja en el gráfico de la figura 7.5.

Cada extremo mantiene una cuenta de los paquetes LQR. tramas y bytes que ha transmitido (contadores *Out*). Así mismo puede consultar las estadísticas de recepción del interfaz del dispositivo (datos *SaveIn*) que reflejan los paquetes LQR, tramas y bytes que ha recibido correctamente, así como los descartes y los errores. Por su parte, los paquetes LQR que recibe un extremo constan de

los últimos contadores de transmisión que él mismo envió (*LastOut*),

las estadísticas de recepción del otro extremo (*PeerIn*) y

los contadores de transmisión del otro extremos.

La formación de un paquete LQR se muestra en la figura 7.5 y consiste en

copiar los contadores *PeerOut* del paquete LQR recibido como contadores *LastOut* para el paquete LQR que se vaya a enviar,

copiar las estadísticas de recepción *SaveIn* como contadores *PeerIn* para el paquete LQR que se vaya a enviar, y

copiar los contadores *Out* como contadores *PeerOut* para el paquete LQR que se vaya a enviar.

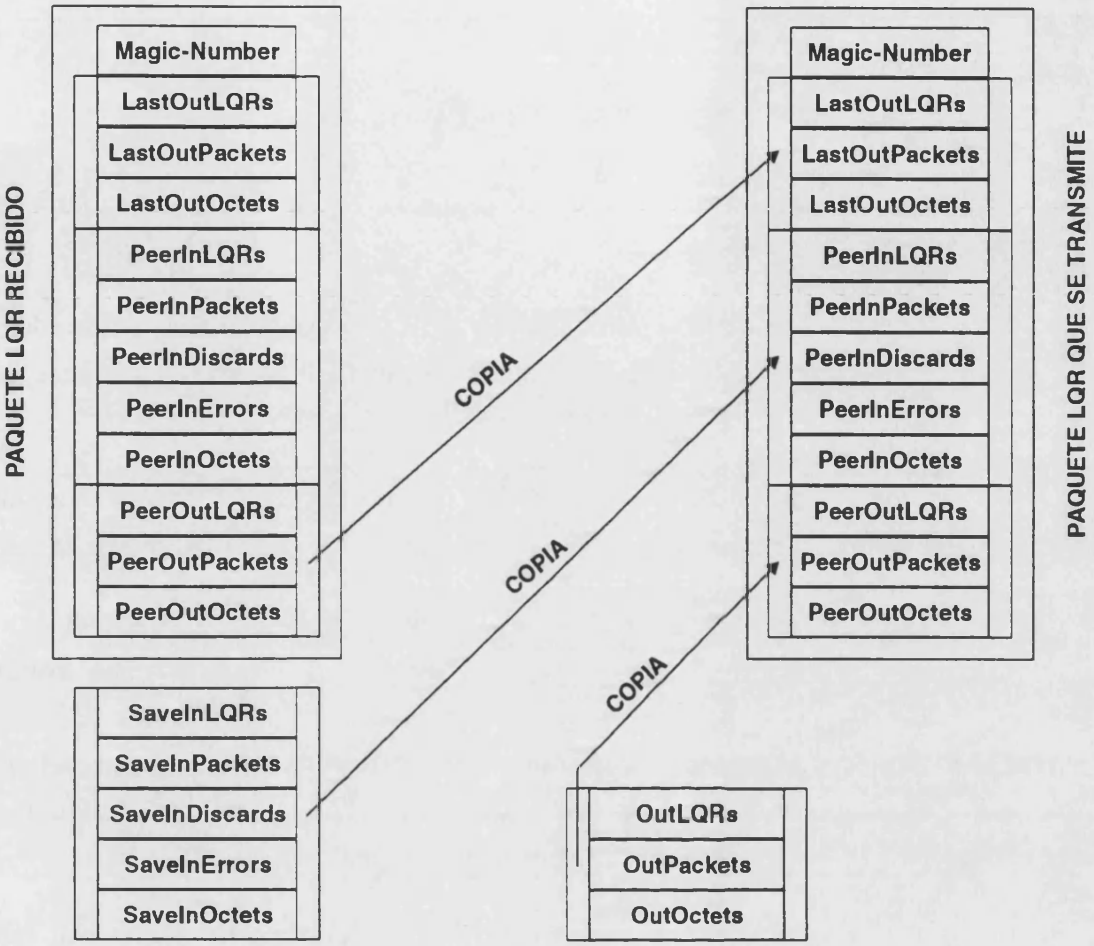


Fig. 7.5: *Protocolo LQP: paquetes y contadores*

El código fuente del *daemon* pppd debe ser modificado (anexo A.2.1) para aceptar una nueva opción 'tlqp' que, acompañada de una indicación de intervalo temporal, activa el protocolo LQP. Una vez la opción de protocolo LQP ha sido negociada con éxito, la implementación envía paquetes LQR a intervalos periódicos indicados en la negociación (lo que llamaremos modo activo de monitorización.

Si el intervalo negociado es cero, no se envían paquetes LQR periódicamente, sino únicamente como respuesta a la recepción de otro paquete LQR (lo que llamaremos modo pasivo). Las implementaciones en los extremos de un enlace PPP pueden enviar ambas en modo activo, con intervalos independientes, o una en modo activo y otra en modo pasivo.

En cualquier caso, el intervalo entre paquetes LQR debe ser suficiente para interferir mínimamente en el tráfico activo.

7.3.4.2. UTILIZACIÓN DE LOS DATOS DE CALIDAD

Cada vez que se recibe un paquete LQR, los datos contenidos se utilizan para determinar la calidad del enlace:

- los campos *LastOut* (la última cuenta de lo que envió) son comparados con los campos *PeerIn* (lo que el otro extremo recibió correctamente o los errores que detectó) para determinar la calidad del enlace saliente
- los contadores *SaveIn* (la estadística de recepciones) son comparados con los campos *PeerOut* (lo que el otro extremo envió) para determinar la calidad del enlace entrante.

Cuando hay pérdida de datos en alguno de los dos sentidos, debe continuar monitorizándose el enlace de igual modo. Intervalos excesivamente cortos entre LQRs permite detectar picos en los errores aunque a costa de disminuir la eficiencia del enlace para el tráfico activo; intervalos excesivamente largos entre LQRs prolongan innecesariamente la latencia de un fallo permanente.

Las decisiones adoptadas por una implementación de LQP usando enlaces con calidad deficiente pueden ser distintas, como por ejemplo la indicación a un *router* para que dé preferencia a encaminamientos alternativos o la desconexión del enlace y el establecimiento de un enlace alternativo

En nuestra implementación, los datos de monitorización de cada uno de los enlaces simples se disponen en una zona de memoria compartida a la que accede un programa de usuario, 'monitor.c' (ver anexo A.3.3), que obtiene datos conjuntos del estado de ambos enlaces.

Los datos de la monitorización conjunta deben conducir a la selección de un modo de funcionamiento del interfaz MLP correspondiente a uno de los modelos de gestión de la redundancia, tal como se explicó en el apartado 5.5.1. En el anexo A.3.4 se muestra el código de un programa de usuario para que el operador seleccione el modo de funcionamiento. La definición de una política de reconfiguración concreta, especificando exactamente para qué circunstancias debe producirse una transición entre modelos

(refiérase nuevamente al apartado 5.5.1) permitirá realizar un simple programa que sin intervención de operador cambie el modo de funcionamiento de MLP según las circunstancias.

7.3.5. CARACTERÍSTICAS DE LA ORGANIZACIÓN PROPUESTA

Esta organización presenta tres características principales:

el código del *driver* MLP es independiente del hardware, dado que la gestión del hardware la hacen los *drivers* PPP ya presentes en el sistema. Por tanto el *driver* con la arquitectura descrita es transportable a una máquina con hardware diferente.

la modificación requerida del núcleo es mínima y por tanto mínimo el impacto sobre el resto del sistema operativo y de la máquina

el control del modo de funcionamiento del multienlace por un programa de nivel de usuario favorece su uso como plataforma de pruebas ya que la modificación del *daemon* se limita a suspenderlo, modificar el programa y lanzar el nuevo *daemon*, manteniendo la actividad del resto de la máquina. De este modo es sencillo experimentar diversos procedimientos para el control del multienlace y evaluar la fiabilidad y tolerancia alcanzada.

De este modo, el interfaz MLP en combinación con el proceso de monitorización y decisión tiene la función de actuar como capa de recubrimiento de los enlaces simples para dotar al conjunto de la fiabilidad y la tolerancia a fallos requerida cumpliendo una doble función:

actuar de capa de recubrimiento de los recursos individuales, asignando trabajos a dichos recursos y monitorizando su funcionamiento.

intermediar entre dichos recursos y los servicios solicitados por el sistema a través del *driver* o interfaz normalizada

7.4. REALIZACIÓN DE PRUEBAS

La primera implementación del MLPPP ha sido hecha sobre un par de ordenadores PC486 con sistema operativo Linux (versión 1.2.13). Tanto el código del *driver* MLP como el de los programas de usuarios han sido escritos en lenguaje C. Los enlaces simples PPP se han establecido con la versión 2.1.2b de PPP para Linux a través de comunicaciones asíncronas por puertos RS232 y líneas de módem nulo operando a 19200 baudios, con 8 bits de datos, 1 bit de parada y sin paridad.

El sistema ha sido probado inyectando fallos en las líneas de transmisión como se describe en el subapartado siguiente.

7.4.1. SIMULACIÓN DE FALLOS

7.4.1.1. INYECCIÓN DE ERRORES

Para probar el funcionamiento de los algoritmos para tolerancia a fallos en los enlaces, se han inyectado errores por separado en cada uno de los sentidos de los enlaces PPP simples. Para ello se ha utilizado el campo *metric* de la estructura asociada a cada enlace PPP simple. Según el valor de dicho campo, que puede ser alterado por una llamada de control del interfaz de PPP y teniendo sus bits menos significativos el siguiente significado:

0x0000 línea libre de errores,

0x0100 alteración en la próxima trama: se altera un byte cualquiera de la trama antes de enviarla, y después se establece el modo libre de errores,

0x0400 línea cortándose: se envía una fracción de la trama y después se establece el modo 'línea cortada'. La cuenta de salida incluye todos los bytes, enviados o no,

0x0800 línea cortada: no se transmite byte alguno, pero se contabilizan como si se hubieran transmitido,

la función *ppp_kick_tty* que produce el envío de las tramas a través del puerto serie considera los submodos indicados y envía los bytes en consecuencia, simulando errores físicos en la transmisión a través de los canales simples.

7.4.1.2. GENERACIÓN DE FALLOS TRANSITORIOS

Los fallos transitorios se simulan introduciendo errores mediante el programa 'berm' (berm.c recogido en el anexo A) inyectando errores a intervalos aleatorios generados a partir de la velocidad de transmisión ('speed') y la tasa de errores como probabilidad de que un bit sea alterado ('ber').

Los intervalos entre errores se determinan por un valor aleatorio entre 0 y 2/(ber*speed), por tanto un tiempo medio entre errores igual a 1/(ber*speed), lo cual supone aproximadamente un error cada 1/ber bits transmitidos (si el canal estuviera plenamente ocupado). Al transcurrir el intervalo así determinado, se inyecta un error mediante el submodo 'alteración en la próxima trama'.

7.4.2. EXPERIENCIAS CON FALLOS PERMANENTES

Las experiencias con fallos permanentes se han realizado simulando éstos mediante el submodo 'línea cortada' antes descrito y utilizando la orden *ping* para comprobar el funcionamiento de la comunicación entre los dos nodos unidos por el punto a punto.

La simulación de un fallo permanente en un enlace PPP aislado o en uno de los enlaces que forman parte de un multienlace PPP gestionado según el modelo M6 supone la pérdida de todos los paquetes transmitidos posteriormente a la aparición del error y la reanudación de la transmisión correcta tan pronto se restaura el enlace averiado.

Por su parte, la simulación de un fallo permanente en sólo uno de los enlaces PPP de un multienlace PPP gestionado según el modelo M4 (repuestos activos; transmisiones duplicadas) no produce ningún efecto apreciable por el nivel de red ya que los fallos son enmascarados. La simulación simultánea de fallos permanentes en ambos enlaces supone evidentemente la pérdida de todos los paquetes, que empiezan a llegar correctamente tan pronto se restaura uno cualquiera de los enlaces averiados.

La implementación del modelo M4 para multienlaces que hemos realizado cumple en consecuencia las especificaciones expuestas en presencia de cortes individuales cubriendo satisfactoriamente los objetivos cualitativos de tolerancia a fallos permanentes de este modelo.

7.4.3. EXPERIENCIAS CON ERRORES TRANSITORIOS

Se han realizado una serie de experiencias con inyección de errores transitorios para comprobar el funcionamiento de los modelos de gestión M4 (apartado 5.4.5.) y M6 (apartado 5.4.7) para multienlaces PPP, comparándolos entre sí y respecto a los resultados obtenidos para un enlace PPP simple.

Para ello se han planteado dos tipos de experiencias: resultados obtenidos con la orden *ping* y resultados obtenidos con la orden ftp, y ambas experiencias realizadas inyectando errores con tres tasas BER diferentes (2e-5, 1e-5 y 5e-6).

La orden *PING*: usa el protocolo ICMP de control del nivel de red para enviar repetidamente paquetes *ECHO_REQUEST* de un determinado tamaño (utilizamos el valor por defecto de 64 bytes) a otro nodo, provocando en aquel una respuesta *ECHO_RESPONSE*. La aplicación determina el tiempo transcurrido hasta recibir completamente el eco de respuesta (*round-trip*). En caso de perderse un paquete (en la ida o en la vuelta), no se solicita retransmisión, y la aplicación contabiliza el número de paquetes perdidos. Las experiencias realizadas han consistido en 5 series de 30000 paquetes para cada modelo y tasa de errores. Los resultados obtenidos con *ping* permiten medir la eficiencia directa del enlace punto a punto para la transmisiones de bloques cortos de datos sin utilizar retransmisiones, obteniendo el tiempo medio empleado por solicitud. El porcentaje de pérdidas se determina a partir las estadísticas de las interfaces PPP (para medir las pérdidas en un enlace simple) o MLP (para los multienlaces PPP), tomando el número de paquetes salientes del extremo emisor y el número de paquetes entrantes del extremo receptor. Se observa que los resultados así obtenidos concuerdan con el porcentaje de pérdidas dado por *ping*.

La aplicación *FTP* utiliza el protocolo de transferencia de ficheros sobre TCP/IP utilizando retransmisiones gestionadas por el nivel de transporte cuando se pierden paquetes. Estableciendo una transferencia de un fichero entre los dos nodos unidos por el multienlace, el fichero es fragmentado y entregado al nivel de enlace de datos en paquetes cuyo tamaño máximo viene determinado por la MRU de PPP o de MLPPP (utilizamos el valor por defecto de 1500 bytes). La experiencia ha consistido en realizar 5 veces la transferencia de un fichero de 2.000.000 bytes para cada modelo y tasa de errores. *Ftp* proporciona resultados tanto de el tiempo total empleado como de la eficiencia en KBytes por segundo. Para medir el porcentaje de pérdidas de paquetes de 1500 bytes utilizamos nuevamente las estadísticas de las interfaces PPP o MLP. Estos resultados nos permitirán evaluar tanto el porcentaje de pérdidas para paquetes largos como la eficiencia en la

transmisión de bloques largos de datos con uso de retransmisiones por encima del nivel de enlace de datos.

En cada uno de los enlaces PPP simples se ha empleado campo de protocolo de un byte, compresión (supresión) del campo de dirección y de control y tramas sin numerar. Para el protocolo MLPPP se han empleado números de secuencia cortos (de 12 bits; módulo 4096). Se han utilizado siempre Secuencias de Verificación de Tramas (SVT o FCS) cortas (de 16 bits).

7.5. RESULTADOS ANTE ERRORES TRANSITORIOS

7.5.1. PING (PAQUETES DE 64 BYTES)

		stats	ping
ber	**tipo**	**% perd.**	**ms/trama**
0.000020	PPP	1,4	125,7
0.000010	PPP	0,7	
0.000005	PPP	0,4	
0.000020	MP4	0,0	128,2
0.000010	MP4	0,0	
0.000005	MP4	0,0	
0.000020	MP6	1,5	95,0
0.000010	MP6	0,7	
0.000005	MP6	0,4	

% perd.	**MP4**	**PPP**	**MP6**
2*E-5	0,0	1,4	1,5
1*E-5	0,0	0,7	0,7
5*E-6	0,0	0,4	0,4

ms/trama	**MP4**	**PPP**	**MP6**
media	128,2	125,7	95,0

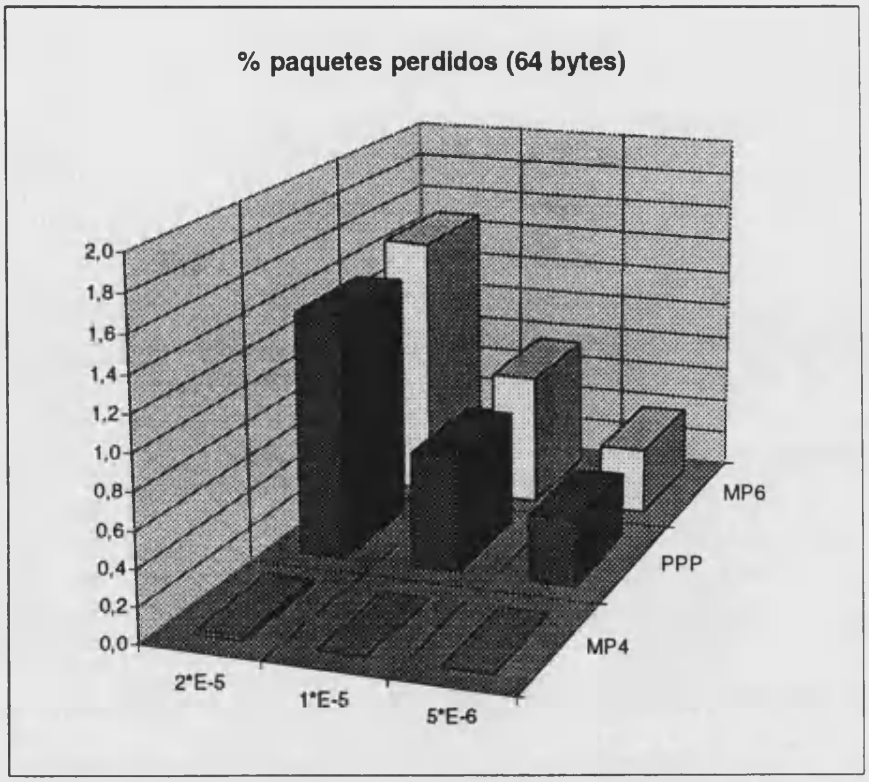


Fig 7.6. *Paquetes cortos perdidos*

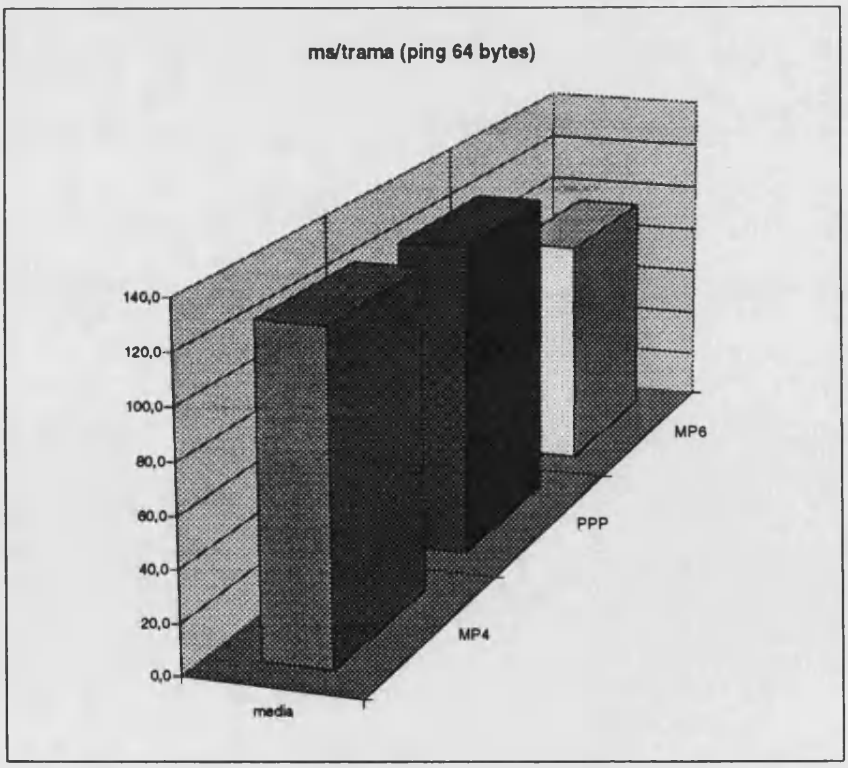


Fig 7.7. *Tiempo de viaje de paquetes cortos*

7.5.2. FTP (CON MRU=1500)

		stats	ftp
ber	**tipo**	**% perd.**	**KBytes/s**
0.000020	PPP	27,7	1,19
0.000010	PPP	14,7	1,41
0.000005	PPP	7,5	1,53
0.000020	MP4	7,1	1,53
0.000010	MP4	2,2	1,61
0.000005	MP4	0,6	1,64
0.000020	MP6	27,1	2,24
0.000010	MP6	14,7	2,62
0.000005	MP6	7,4	2,85

% perd.	**MP4**	**PPP**	**MP6**
2*E-5	7,1	27,7	27,1
1*E-5	2,2	14,7	14,7
5*E-6	0,6	7,5	7,4

KBytes/s	**MP4**	**PPP**	**MP6**
2*E-5	1,53	1,19	2,24
1*E-5	1,61	1,41	2,62
5*E-6	1,64	1,53	2,85

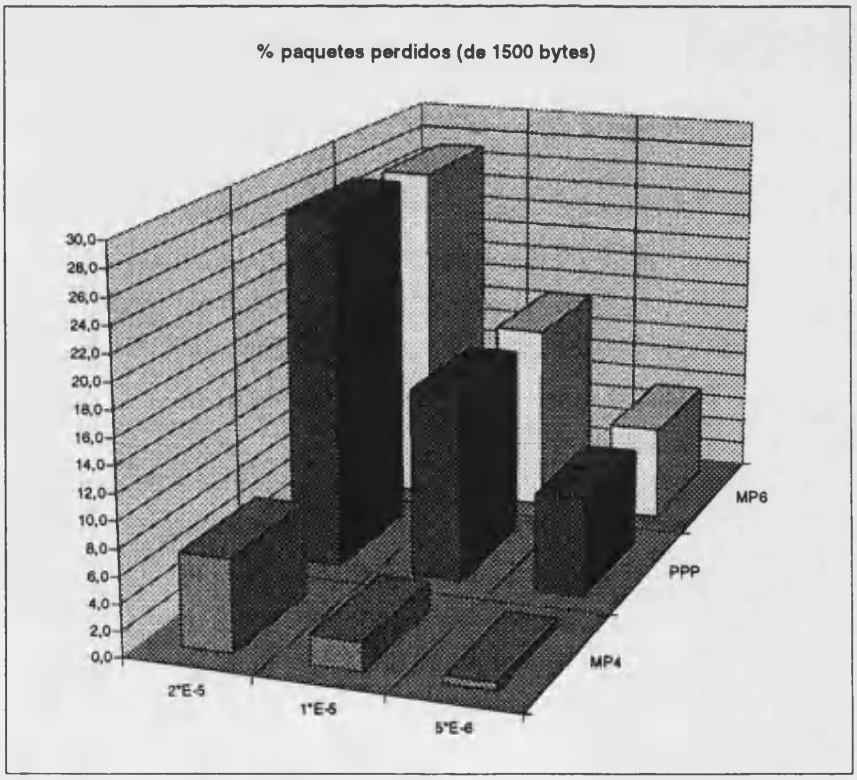


Fig 7.8. *Paquetes largos perdidos*

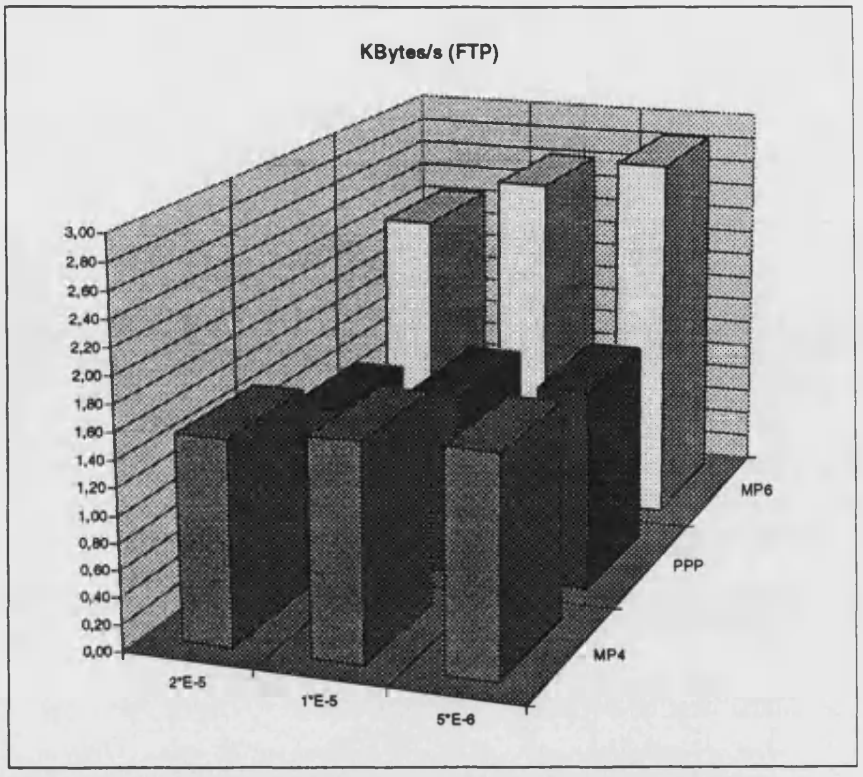


Fig 7.9. *Eficiencia en la transmisión para FTP*

7.5.3. INTERPRETACIÓN DE LOS RESULTADOS

Las figuras 7.6 y 7.8 muestran como, independientemente de la tasa de errores, el porcentaje de paquetes perdidos en el nivel de enlace de datos para un enlace simple PPP y para un multienlace PPP según el modelo M6 es muy similar, siendo significativamente menor el porcentaje de pérdidas para el multienlace PPP gestionado con el modelo M4. Por otra parte, el uso de paquetes más largos supone para todos los modelos un incremento considerable del porcentaje de pérdidas.

La figura 7.7. muestra como el tiempo necesario para la transmisión de paquetes cortos es ligeramente mayor para el multienlace PPP con el modelo de gestión M4 que para un enlace PPP simple, debido a las cabeceras adicionales y al ligero aumento de las instrucciones a ejecutar (las que rigen el protocolo MLPPP). Por su parte el tiempo de viaje para un multienlace PPP según el modelo de gestión M6 es bastante superior a la mitad del valor para enlace PPP simple, siendo ello debido a que el tiempo medido incluye también el tiempo de procesamiento en cada extremo, y éste no sólo no se reduce por la mitad sino que aumenta ligeramente al utilizar MLPPP.

Por último, la figura 7.9 muestra como la eficiencia en la transmisión de bloques largos es mayor en un multienlace PPP según el modelo M4 que en un enlace PPP simple, siendo ello debido a que el modelo M4 realiza un número menor de retransmisiones por perder menos tramas. Además, la diferencia entre ambos casos se acentúa conforme crece la tasa de errores en los enlaces simples. En la transmisión de tramas largas, donde el tiempo de transmisión supone una fracción mucho mayor que el tiempo de procesamiento, la eficiencia del modelo de gestión M6 es cercana al doble de la obtenida con los otros dos modelos cuando la tasa de errores es baja, pero decae con el incremento de la tasa de errores acercándose a los resultados del modelo M4.

CONCLUSIONES

El trabajo de investigación presentado en esta memoria, tuvo su origen en la pregunta genérica de saber si la metodología de tolerancia a fallos se podía extender con sus procedimientos propios a problemas de transmisión de datos para ingenierías con requisitos especiales, como podrían ser los requisitos de fiabilidad de los sistemas ITS, sistemas cuyo desarrollo se basa tanto en el hardware como en el software y en las comunicaciones.

El análisis del estadio de evolución de los ITS nos ha permitido constatar la necesidad de una arquitectura ITS que establezca un marco general para el diseño e implementación de estos sistemas, incluyendo la definición de las interfaces y formas de comunicación entre sus subsistemas físicos. Igualmente se ha puesto de manifiesto la necesidad de que estas comunicaciones empleadas en los ITS se adapten a algunos estándares procedentes del campo de las redes telemáticas de propósito general.

El trabajo realizado ha mostrado como dentro de los sistemas de control de tráfico (el área de los ITS que requiere mayor garantía de funcionamiento) existen aplicaciones críticas con requisitos especiales de fiabilidad no completamente resueltos por los sistemas actuales, en particular en lo relativo a la fiabilidad en las comunicaciones. Las posibles soluciones a estos requisitos deben mantener los interfaces definidos en la arquitectura y en los protocolos estandarizados. Una vez obtenida la evidencia de que dentro de la arquitectura de los ITS son necesarios nuevos desarrollos que aseguren la tolerancia a fallos, nos hemos planteado la obtención de procedimientos adecuados basados en técnicas que ya habían demostrado su eficacia en la construcción de sistemas fiables.

Tratando el problema de la fiabilidad de la comunicación punto a punto en el nivel de enlace de datos, a partir del protocolo PPP, estandarizado para Internet, y ya propuesto para los ITS, hemos investigado la utilización del protocolo MLPPP que gestiona múltiples enlaces y que estando en proceso de difusión para Internet no está planteado como estándar en ITS de acuerdo con la bibliografía disponible. Partiendo de la existencia de más de un

enlace, se plantean diferentes procedimientos para gestionar esta redundancia de recursos. Los resultados obtenidos nos indican que la aplicación de la metodología general de tolerancia a fallos permite trasladar los diferentes modelos de gestión de la redundancia al desarrollo de algoritmos para la gestión de la redundancia de enlaces de comunicación punto a punto, todos ellos emplazados en un nivel de enlace de datos e implementables por software. En este punto nuestra conclusión es que la arquitectura de ITS debe incorporar como estándar el protocolo MLPPP como recientemente ha hecho con el PPP. Con ello resolverá las necesidades de fiabilidad detectadas en los capítulos anteriores.

El desarrollo de una ingeniería específica para un caso de estudio de un sistema de control lineal de la velocidad, ha incorporado estos procedimientos de tolerancia a fallos dentro de la arquitectura global propuesta para ITS, y ha mostrado la potencialidad de uso de los mismos para el establecimiento de un sistema de control tolerante a fallos tanto en sus nodos como en sus comunicaciones.

Los experimentos llevados a cabo con el código escrito para implementar estas soluciones han mostrado que el emplazamiento de la gestión para la tolerancia a fallos como parte de las labores a llevar a cabo por el MLPPP permite mantener satisfactoriamente las interfaces del sistema con los niveles superiores, alcanzando el objetivo deseado de que la fiabilidad del elemento "comunicaciones" sea integrable de forma transparente con el resto de elementos de un sistema tolerante o no.

La conclusión de esta investigación es la constatación no sólo de la necesidad de establecer metodológicamente los requisitos de fiabilidad de los sistemas ITS sino de que puedan obtenerse soluciones estandarizables para los mismos. Esta conclusión se instancia en la consideración de que el protocolo MLPPP procedente de Internet, y que presenta unas interfaces iguales a las de otro estándar de nivel de enlace de datos como es PPP, debería ser considerado para su posible adopción como estándar de ITS. Asimismo sería deseable que se propusiera un estándar de modo de funcionamiento del MLPPP tolerante a fallos, en cuyo caso nuestros resultados indican que uno basado en el modelo M4 de esta memoria sería el más indicado.

LÍNEAS ABIERTAS PARA FUTURAS INVESTIGACIONES

Los sistemas ITS constan de tecnologías diferentes según el tipo de información que necesiten procesar o bien difundir entre/para los usuarios. La tecnología que atañe a los servicios de información al usuario, particularmente las ayudas a la conducción, son utilizadas en beneficio de una mejora global del sistema vial. Las ayudas que un conductor puede recibir están básicamente relacionadas con el trayecto o el estado del tráfico. Las primeras permiten al conductor hacer una preparación del trayecto, mientras que las segundas le permiten hacer una adaptación dinámica del trayecto planeado.

En los mecanismos de intercambio de información, uno de los objetivos perseguidos estriba en que ésta adopte estándares basados sobre la arquitectura definida para los ITS, al objeto de permitir la interoperatividad entre distintas fuentes de información empleando una infraestructura telemática de uso común abandonando las prácticas históricas de medios de comunicación privados y generación de protocolos propietarios.

Dichos estándares se plantean desde el punto de vista de los estándares vigentes en el mundo de las comunicaciones de carácter general. Las líneas abiertas por la presente memoria no estriban tanto en el estudio de los medios de transmisión, sino más bien en realizar un estudio detallado y minucioso de aquellos estándares que por sus características técnicas están evolucionando con gran éxito, como es el caso de los estándares NTCIP americanos que integran dentro de su arquitectura los protocolos TCP y/o UDP de transporte sobre redes IP, sobre los cuales se pretende construir todos aquellos servicios de ayuda de al conductor.

Para fijar los protocolos y las interrelaciones que puedan existir entre todos los elementos del sistema telemático que dará soporte al conductor, se establecen además de los protocolos de transporte y de red, la necesidad de estandarizar las interfaces gráficas de aplicaciones (posición planteada en la reunión de Nov'96 de 26 empresas del sector en Bruselas) para todas las aplicaciones que vayan a ejecutarse desde un automóvil, de

modo que la flexibilidad de diferentes implementaciones que distintos fabricantes realicen quede reducida por la necesaria compatibilidad entre ellas.

Sobre los estándares definidos, las características de los protocolos TCP/IP elegidos, sobre los cuales se tiene acceso universal, permiten aprovechar con facilidad las infraestructuras creadas así como los servicios existentes para el intercambio de información y, como no, ayuda a la conducción. La tecnología Internet basada sobre IP ha demostrado ser el marco ideal para la provisión de servicios al gran público básicamente debido a que:

- IP es una tecnología puramente de red independiente de la infraestructura física (ATM, IEEE 802.3, IEEE 802.4, Spread-Spectrum, GSM, DAB ...) permitiendo multitud de medios de acceso de diferente coste, capacidad y prestaciones;

- existen numerosas aplicaciones y software que explotan su tecnología (librerías, rutinas de bajo nivel ...) que permiten fácilmente su reutilización y rápida generación de aplicaciones basadas en red;

- dispone de una red de direcciones global desde su planteamiento con un espacio de nombres asociados que son transformables dinámicamente (*Domain Name Servers* DNS).

Los resultados de la investigación abonan la posibilidad de utilizar todas estas tecnologías en ITS y señalar sus estándares como válidos para problemas relacionados con el transporte.

Por otro lado, estudios estadísticos y de mercado revelan la gran demanda de equipos y sistemas informáticos (computadores de a bordo) para equipar los automóviles, con la finalidad de dar soporte y ayuda en la conducción, además de facilitar aplicaciones ofimáticas de uso común. Dichos computadores no han de descuidar el modo en que el conductor haga uso de dichas aplicaciones, lógicamente en manos libres. Como muestra del citado interés destaca el desarrollo de un nuevo sistema operativo (Microsoft Windows CE), que permite esta operación de manos libres, para portátiles y ordenadores de a bordo.

Las líneas de investigación análogas a la desarrollada en esta memoria tendrán como objetivo incluir, como información de valor añadido, aplicaciones multimedia basadas

sobre TCP/IP usando la infraestructura de Internet para dar servicio al conductor. La infraestructura que se está generando para ayuda a la conducción a través de los servicios telemáticos, permite incorporar tecnologías emergentes junto a las ya consolidadas, adaptadas a las necesidades planteadas en los problemas de conducción.

Puntos clave de dicha investigación basándonos en la arquitectura ITS son:

- adquisición de la información en los subsistemas externos para monitorizar aquellos puntos clave como túneles, accesos a ciudades, cruces conflictivos ...

- producción en el sistema central de información para los usuarios.

- técnicas y protocolos para transmisión al subsistema del vehículo de la información de ayuda al conductor, basándose en las prestaciones de las redes TCP/IP.

La información procesada que ha de facilitarse al conductor podría usar la videoconferencia basada en *multicast*, facilitada por los recientes protocolos IPv6, 6BONE y en menor medida IPv4 con MBONE. Dicha tecnología (una migración de las técnicas utilizadas para la Red Digital de Servicios Integrados -RDSI- pero adaptadas a arquitecturas no deterministas como es Internet) sustituye o recoge la percepción humana del hecho observado para poder reproducirla de manera controlada en otro lugar, permitiendo al conductor ser espectador de todo aquello que acontece en el lugar donde se monitoriza. La videoconferencia, como aplicación multimedia, permite integrar información de diferentes fuentes bajo un mismo acceso, pudiendo con ello vigilar el estado de las carreteras, con especial interés en aquellos lugares que por sus condiciones son poco saludables e inhóspitos (por ejemplo el caso de túneles), accediendo a información tal como iluminación, calidad del aire, sonido, vídeo ...

Una propuesta de trabajo de esta investigación sería, a grandes rasgos [Fel97]:

- Planteamiento y exposición del problema a resolver en tráfico. Soluciones posibles al problema y definición clara de los objetivos a conseguir en la investigación. Planteamiento de tratamiento de información (información masiva de gran volumen de datos) para transmisión de videoconferencia bajo redes TCP/IP.

- Evaluación de las investigaciones realizadas sobre el problema planteado. Estudio y análisis de las soluciones realizadas para otras disciplinas. Consideraciones realizadas y estándares definidos para transmisión de videoconferencia sobre RDSI.

- Requisitos de sistemas multimedia distribuidos. Estudio de la información a transmitir y procesar como requisitos de ayuda al conductor. Valor añadido a la información transmitida desde los subsistemas centrales.

- Estudio de protocolos TCP/IP (Internet) y Mbone. Estudio de las características *multicast* incorporadas al protocolo IP y evaluación de dichos mecanismos sobre versión 6 (IPv6) considerando la repercusión de tiempo real inherente a los servicios de videoconferencia sobre estos protocolos.

- Compresión de la información. Algoritmos adaptativos al ancho de banda del canal, ponderando factor de compresión, velocidad de transmisión dependiente de la saturación del canal y velocidad de procesamiento.

- Modelado del sistema telemático. Diseño de modelos matemáticos basados en teoría de colas o Markovianos que permitan estimar la evolución de sistemas de videoconferencia aplicados sobre redes TCP/IP como soporte de información al conductor, con variabilidad de carga y saturación en la red debido a sus características no deterministas inherentes. Estimación de los parámetros relevantes del sistema de colas e introducción de datos en dicho sistema procedentes de analizadores software de protocolos.

ANEXO A

CÓDIGO DE LA IMPLEMENTACIÓN

A.1. CÓDIGO ENLAZADO EN EL NÚCLEO

A.1.1. MLP.C

```
/*
   Multilink PPP for Linux

   RFC1717: PPP Multilink
   mlp-2.4
*/

#include <linux/config.h>

#include <linux/kernel.h>
#include <linux/sched.h>
#include <linux/types.h>
#include <linux/fcntl.h>
#include <linux/interrupt.h>
#include <linux/ptrace.h>
#include <linux/ioport.h>
#include <linux/in.h>
#include <linux/malloc.h>
#include <linux/tty.h>
#include <linux/errno.h>
#include <linux/sched.h>   /* to get the struct task_struct */
#include <linux/string.h>  /* used in new tty drivers */
#include <linux/signal.h>  /* used in new tty drivers */
#include <asm/system.h>
#include <asm/bitops.h>
#include <asm/segment.h>

#include <linux/netdevice.h>
#include <linux/skbuff.h>
#include <linux/inet.h>

#include <linux/ppp.h>

#include <linux/ip.h>
```

```
#include <linux/tcp.h>

#include <linux/if_arp.h>
#ifndef ARPHRD_PPP
#define ARPHRD_PPP 0
#endif

#include <linux/mlp.h>

  /* Prototipos */
int mlp_init(struct device*);
static int mlp_open(struct device*);
static int mlp_close(struct device*);
static int mlp_xmit(struct sk_buff*,struct device*);
extern int ppp_xmit(struct sk_buff*,struct device*);
static struct enet_statistics *mlp_get_stats(struct device*);
static int mlp_header(unsigned char*,struct device*,
  unsigned short,void*,void*,unsigned,struct sk_buff*);
static unsigned short mlp_type_trans(struct sk_buff*,struct device*);
static int mlp_rebuild_header(void*,struct device*,unsigned long,struct
sk_buff*);
static int mlp_ioctl(struct device*,struct ifreq*,int);
int mlp_recep(unsigned char*,int,int,struct device*);

/*=========================================================*/

int mlp_init(struct device *dev)
  {
  static int primera_vez=1;
  int i;

  if ( primera_vez!=0 )
    {
    primera_vez=0;
    printk(KERN_INFO ">>> *** MLP version 2.4 *** <<<\n");
    }
    /* device INFO: Funciones */
  dev->open            = mlp_open;
  dev->stop            = mlp_close;
  dev->hard_start_xmit = mlp_xmit;
  dev->get_stats       = mlp_get_stats;
  dev->hard_header     = mlp_header;
  dev->type_trans      = mlp_type_trans;
  dev->rebuild_header  = mlp_rebuild_header;
  dev->do_ioctl        = mlp_ioctl;
    /* device INFO: Variables */
  dev->mtu             = PPP_MTU;
  dev->hard_header_len = 0;
  dev->addr_len        = 0;
  dev->type            = ARPHRD_PPP;
  dev->flags           = IFF_POINTOPOINT;
  dev->family          = AF_INET;
  dev->pa_addr         = 0;
  dev->pa_brdaddr      = 0;
  dev->pa_mask         = 0;
  dev->pa_alen         = sizeof(unsigned long);
```

```
  dev->metric           = TOLM_P0; /* no se usa en esta version de Linux
*/
          /* aqui se usa para el modo de tolerancia a fallos utilizado
*/
    /* Inicializa lista de buffers skb */
  for (i=0; i<DEV_NUMBUFFS; i++)
    skb_queue_head_init(&dev->buffs[i]);
    /* Inicializa estructura privada para estadisticas */
  dev->priv = kmalloc(sizeof(struct enet_statistics),GFP_KERNEL);
  memset(dev->priv,0,sizeof(struct enet_statistics));
  return(0);
  }

/*=========================================================*/

static int mlp_open(struct device *dev)
  {
  return(0);
  }

/*=========================================================*/

static int mlp_close(struct device *dev)
  {
  return(0);
  }

/*=========================================================*/

static struct enet_statistics *mlp_get_stats(struct device *dev)
  {
  struct enet_statistics *stats;

  stats = (struct enet_statistics*)dev->priv;
  return(stats);
  }

/*=== TRANSMISION ========================================*/

static int mlp_xmit(struct sk_buff *skb, struct device *dev)
  {
  static int frame_out=0;
  struct sk_buff *skb2; /*para duplicados o divididos*/
  unsigned char *ptr;   /*para duplicados o divididos*/
  int llarg;
  struct enet_statistics *stats;
  struct device *ppp_dev;
  int ret1=0,ret2=0;

  if ( (dev->flags & IFF_UP)==0 )
    {
   printk(KERN_WARNING "mlp_xmit: interface %s inactivo para IP\n",dev-
>name);
    dev_kfree_skb(skb,FREE_WRITE);
    return(0);
    }
```

```
stats = (struct enet_statistics *)dev->priv;

if(dev->metric==TOLM_DUP) { /* modo ENVIO DUPLICADO */
  skb2 = alloc_skb(skb->len,GFP_ATOMIC);
  if(skb2==NULL) {
    printk("\nMLP: no hay memoria para crear una nueva trama\n");
    return(-1);
    }
  ptr = (unsigned char *)(skb2+1);
  skb2->free = 1;
  skb2->len = skb->len;
  skb2->h.raw = ptr;
  memcpy(ptr,(unsigned char*)(skb+1), skb->len);

  skb->pkt_type = PROTO_MLP;
     skb->saddr = FULL_FRAG + TOLM_COP_0 + frame_out;
  ppp_dev = dev_get("ppp0");
  ret1 = ppp_dev->hard_start_xmit(skb,ppp_dev);
     /* no se guarda nada, pues no incluye retransmisiones */
  dev_kfree_skb(skb,FREE_WRITE);

  skb2->pkt_type = PROTO_MLP;
  skb2->saddr = FULL_FRAG + TOLM_COP_1 + frame_out;
  ppp_dev = dev_get("ppp1");
  ret2 = ppp_dev->hard_start_xmit(skb2,ppp_dev);
     /* no se guarda nada */
  dev_kfree_skb(skb2,FREE_WRITE);

  stats->tx_packets++;

  if(frame_out==MASK_SEQ)
   frame_out=0;
  else
   frame_out++;
  return(ret1+ret2);
  }

else if(dev->metric==TOLM_DIV) { /* modo ENVIO DIVIDIDO */
     /* se van a enviar dos tramas. Para que sean n y n+1 */
  if(frame_out==MASK_SEQ)
   frame_out=0;
  else
      frame_out++;

     llarg = skb->len;
     skb2 = alloc_skb(llarg/2,GFP_ATOMIC);
  if(skb2==NULL) {
    printk("\nMLP: no hay memoria para crear una nueva trama\n");
    return(-1);
    }
  ptr = (unsigned char *)(skb2+1);
  skb2->free = 1;
     skb2->len = llarg/2;
  skb2->h.raw = ptr;
     /* Copia en skb2 la segunda mitad del paquete */
     memcpy(ptr,((unsigned char*)(skb+1))+((llarg+1)/2), llarg/2);
```

```
      /* Considera ahora solo la primera mitad del paquete */
      skb->len = (llarg+1)/2;

   skb->pkt_type = PROTO_MLP;
      skb->saddr = BEG_FRAG + frame_out;
   ppp_dev = dev_get("ppp0");
      ret1 = ppp_dev->hard_start_xmit(skb,ppp_dev);

      skb->len = llarg; /* restaura el tamanyo original */
      /* no se guarda nada, pues no incluye retransmisiones */
   dev_kfree_skb(skb,FREE_WRITE);

   skb2->pkt_type = PROTO_MLP;
      skb2->saddr = END_FRAG + frame_out + 1;
   ppp_dev = dev_get("ppp1");
      ret2 = ppp_dev->hard_start_xmit(skb2,ppp_dev);

      /* no se guarda nada, pues no incluye retransmisiones */
   dev_kfree_skb(skb2,FREE_WRITE);

   stats->tx_packets++;

      /* se han enviado dos tramas*/
      /* por el ajuste inicial, frame_out estaba entre 0 y MASK_SEQ-1 */
      if(frame_out==(MASK_SEQ-1))
    frame_out=0;
   else
       frame_out+=2;

      return(ret1+ret2);
   }

else if(dev->metric==TOLM_P1) /* ENVIO POR PPP1 */
   ppp_dev = dev_get("ppp1");
else if(dev->metric==TOLM_P0) /* ENVIO POR PPP0 (por defecto) */
   ppp_dev = dev_get("ppp0");
else { /* modo desconocido */
   dev_kfree_skb(skb,FREE_WRITE);
   return(0);
   }

/* si ha llegado hasta aqui, es el modo TOLM_P0 o TOLM_P1 */
skb->pkt_type = PROTO_MLP;
skb->saddr = FULL_FRAG + frame_out;
ret1 = ppp_dev->hard_start_xmit(skb,ppp_dev);
/* aqui no se guarda nada, pues no incluye retransmisiones */
dev_kfree_skb(skb,FREE_WRITE);

/* Actualiza estadisticas */
stats->tx_packets++;

if(frame_out==MASK_SEQ)
   frame_out=0;
else
   frame_out++;

return(ret1);
```

```
  }

/*=== RECEPCION ==========================================*/

int mlp_recep(unsigned char *buff, int len, int flags, struct device
*dev)
  {
  static int rec0=MASK_SEQ;
  static int rec1=MASK_SEQ;
  static int exp=0;
  static unsigned char *mlpbuff=NULL; /* guarda un fragmento */
  static int mlpcount=0;
  int last,frame_in;
  int ret;
  struct device *devm;
  char nom_dev[8];
  struct enet_statistics *stats;
  unsigned char *p;
  int cuen;
  u_short mlpinfo;

  if ( (dev->flags & IFF_UP)==0 )
    {
    printk(KERN_WARNING "mlp_recep: interface %s inactivo para IP\n",dev-
>name);
    return(1);
    }

  /* procesa la cabecera MP (2 bytes fijos) */
  mlpinfo = ((u_short)buff[0])<<8 | (u_short)buff[1];
  frame_in = mlpinfo & MASK_SEQ; /* numero de secuencia */

  if( (mlpinfo&FULL_FRAG) != FULL_FRAG ) { /* paquete dividido */
       if(dev->base_addr == 1) {
            last=rec1;
            rec1=frame_in;
            }
       else {
            last=rec0;
            rec0=frame_in;
            }
       if(frame_in==last) {
            printk(KERN_WARNING "\nrepetido!\n");
            return 0;
            }

       if((1+rec0)!=rec1) { /* dos fragmentos del mismo paquete. Juntar
*/
            if(mlpinfo&BEG_FRAG) { /* poner mlpbuff detras de buff */
              memcpy(buff+len,mlpbuff,mlpcount);
              p=buff+2;
              cuen=len-2+mlpcount;
              }
            else { /* poner buff detras de mlpbuff */
              memcpy(mlpbuff+mlpcount,buff+2,len-2);
              p=mlpbuff;
              cuen=len-2+mlpcount;
```

```
            }
          }
      else if((last<=frame_in && frame_in<exp) ||
           (exp<last && last<=frame_in) || (frame_in<exp && exp<last)){
           /* medio paquete retrasado */
          printk(KERN_WARNING "\nretrasado!\n");
          return 0;
          }
      else { /* medio paquete bueno. Guardar y salir */
          if(mlpbuff==NULL)
            if((mlpbuff=kmalloc(1500,GFP_ATOMIC))==NULL) {
                printk("mlp: fallo reservando un buffer\n");
                return 0;
                }
          mlpcount=len-2;
          memcpy(mlpbuff,buff+2,mlpcount);
          exp=frame_in;
          return 0;
          }
      }
  else {
    p=buff+2;
    cuen=len-2;
    }

  if( mlpinfo&TOLM_COPIA ) { /* trama replicada */
    if( dev->base_addr == 1) {
      last=rec1;
      rec1=frame_in;
      }
    else {
      last=rec0;
      rec0=frame_in;
      }
       /* si, ciclicamente, last<=frame_in<exp es redundante o repetido*/
       if((last<=frame_in && frame_in<exp) ||
            (exp<last && last<=frame_in) || (frame_in<exp && exp<last))
{
          printk(KERN_WARNING "\nredundante\n");
          return 0;
          }
     }

  stats = (struct enet_statistics*)dev->priv;
  stats->rx_packets++;

  /* esperar ahora la siguiente a la ultima aceptada */
  if(frame_in==MASK_SEQ)
    exp=0;
  else
    exp = frame_in+1;

  sprintf(nom_dev,"mlp%ld",(dev->base_addr)/2);
  devm=dev_get(nom_dev);

  ret=dev_rint(p,cuen,flags,devm);
  return 1;
```

```
  }

/*=== CONTROL ==============================================*/
/* ===
   funcion mlp_ioctl asignada a dev->do_ioctl para opciones
   adicionales a las normales, cuyos valores
   desde SIOCDEVPRIVATE hasta SIOCDEVPRIVATE+0xf (inclusive)
   SIOCDEVPRIVATE esta definida en <linux/sockios.h>

   Aqui, el control del modo de operacion tolerante se va a guardar
   en la metrica dev->metric
=== */

static int mlp_ioctl(struct device *dev, struct ifreq *ifr, int cmd)
  {
  int modo;

  switch(cmd) {
    case SIOCGTOLM: /* obtiene (en metric) el modo de tolerancia */
      ifr->ifr_metric = dev->metric;
      return(0);
    case SIOCSTOLM: /* establece (con metric) el modo (si es valido) */
      modo = ifr->ifr_metric;
      if((modo==TOLM_P0)||(modo==TOLM_P1)
          ||(modo==TOLM_DIV)||(modo==TOLM_DUP)) {
        dev->metric = ifr->ifr_metric;
        return(0);
        }
      else
        return(-EINVAL);
    }
  return(-EINVAL);
  }

/*=== FUNCIONES QUE NO SIRVEN PARA NADA ===*/

static int mlp_header(unsigned char *buff, struct device *dev,
  unsigned short type, void *daddr, void *saddr,
  unsigned len, struct sk_buff *skb)
  {
  struct device *ppp_dev;
  int ret;

  if ( (dev->flags & IFF_UP)==0 )
    {
    printk(KERN_WARNING "mlp_header: interface %s inactivo para
IP\n",dev->name);
    dev_kfree_skb(skb,FREE_WRITE);
    return(0);
    }
  return(0);
  }

static int mlp_rebuild_header(void *nulo, struct device *dev,
  unsigned long len, struct sk_buff *skb)
  {
  struct device *ppp_dev;
```

```
  int ret;

  if ( (dev->flags & IFF_UP)==0 )
    {
    printk(KERN_WARNING "mlp_rebuild_header: interface %s inactivo para
IP\n",dev->name);
    dev_kfree_skb(skb,FREE_WRITE);
    return(0);
    }
  return(0);
  }

static unsigned short mlp_type_trans(struct sk_buff *skb, struct device
*dev)
  {
  struct device *ppp_dev;
  int ret;

  if ( (dev->flags & IFF_UP)==0 )
    {
    printk(KERN_WARNING "mlp_type_trans: interface %s inactivo para
IP\n",dev->name);
    dev_kfree_skb(skb,FREE_WRITE);
    return(0);
    }
  return(htons(ETH_P_IP));
  }
```

A.1.2. MLP.H

```
/*
   Include file for Multilink PPP for Linux
*/
#ifndef SIOCDEVPRIVATE
#include <linux/sockios.h>
#endif
#define SIOCGTOLM  (SIOCDEVPRIVATE)
#define SIOCSTOLM  (SIOCDEVPRIVATE+1)

#ifndef PROTO_MLP
#define PROTO_MLP 0x003d
#endif

#define BEG_FRAG  0x8000
#define END_FRAG  0x4000
#define FULL_FRAG 0xC000

#define MASK_PPP  0x3000
#define MASK_SEQ  0x0fff

#define TOLM_COPIA 0x3000
#define TOLM_COP_0 0x1000
#define TOLM_COP_1 0x2000

#define TOLM_P0    1
```

```
#define TOLM_P1    2
#define TOLM_DIV   7
#define TOLM_DUP  11
```

A.1.3. PPP.C (SÓLO MODIFICACIONES)

```
/*
   PPP for Linux
*/

/*
   Sources:

   slip.c

   RFC1331: The Point-to-Point Protocol (PPP) for the Transmission of
   Multi-protocol Datagrams over Point-to-Point Links

   RFC1332: IPCP
            ====================================
   ppp-2.4  = MODIFICADO PARA TRABAJAR CON MLP =
            ====================================
   MODIFICADO PARA ERRORES
   0x100 ERROR AISLADO UNICO
   0x200 PAQUETES ERRONEOS
   0x400 LINEA CORTANDOSE
   0x800 LINEA CORTADA
*/

#ifndef PROTO_MLP
#define PROTO_MLP 0x003d
#endif


static void
ppp_kick_tty (struct ppp *ppp)
{
  register int count = ppp->xhead - ppp->xbuff;
  register int answer;

  ppp->stats.sbytes += count;

/* MODIFICADO */
if( (ppp->dev->metric>=0x100) && (count>0) ) {
  if((ppp->dev->metric) & 0x800)           /* linea cortada */
    count = 0;
  else if((ppp->dev->metric) & 0x400) { /* linea cortandose */
                /* manda 3 por mandar algo, pero incompleto */
    count = 3 ;
    ppp->dev->metric += 0x400;
    }
  else { /* paquete erroneo .Altera el FCS */
    (*(ppp->xhead - 2)) ^= 0xff;
    if( ppp->dev->metric & 0x100 )
```

```
      ppp->dev->metric ^= 0x100; /* se limpia */
    }
  }
/* FIN MODIFICADO */

  answer = tty_write_data (ppp->tty,
                    ppp->xbuff,
                    count,
                    ppp_output_done,
                    (void *) ppp);

  if (answer == 0)
    ppp_output_done (ppp);   /* Should not happen */
  else
    if (answer < 0) {
      ppp->stats.serrors++;
      ppp_output_done (ppp); /* unlock the transmitter */
    }
}


/* on entry, a received frame is in ppp->rbuff
   check it and dispose as appropriate */
static void
ppp_doframe(struct ppp *ppp)
{
  u_char *c = ppp->rbuff;
  u_short proto;
  int count = ppp->rcount;

  /* forget it if we've already noticed an error */
  if (ppp->toss) {
    PRINTKN (1, (KERN_WARNING "ppp_toss: tossing frame, reason = %d\n",
             ppp->toss));
    slhc_toss (ppp->slcomp);
    ppp->stats.rerrors++;
    return;
  }

  /* do this before printing buffer to avoid generating copious output */
  if (count == 0)
    return;

  if (ppp_debug >= 3)
    ppp_print_buffer ("receive frame", c, count, KERNEL_DS);

  if (count < 4) {
    PRINTKN (1,(KERN_WARNING "ppp: got runt ppp frame, %d chars\n",
count));
    slhc_toss (ppp->slcomp);
    ppp->stats.runts++;
    return;
  }

  /* check PPP error detection field */
  if (!ppp_check_fcs(ppp)) {
    PRINTKN (1,(KERN_WARNING "ppp: frame with bad fcs\n"));
```

```
    slhc_toss (ppp->slcomp);
    ppp->stats.rerrors++;
    return;
  }

  count -= 2;                  /* ignore last two characters */

  /* now we have a good frame */
  /* figure out the protocol field */
  if ((c[0] == PPP_ADDRESS) && (c[1] == PPP_CONTROL)) {
    c = c + 2;                 /* ADDR/CTRL not compressed, so skip */
    count -= 2;
  }

  proto = (u_short) *c++;            /* PROTO compressed */
  if (proto & 1) {
    count--;
  } else {
    proto = (proto << 8) | (u_short) *c++; /* PROTO uncompressed */
    count -= 2;
  }

printk(KERN_WARNING "\nprotocolo %x\n",proto);

  /* Send the frame to the network if the ppp device is up */
  if ((ppp->dev->flags & IFF_UP) && ppp_do_ip(ppp, proto, c, count)) {
    ppp->ddinfo.ip_rjiffies = jiffies;
    return;
  }

  /* If we got here, it has to go to a user process doing a read,
     so queue it.

     User process expects to get whole frame (for some reason), so
     use count+2 so as to include FCS field. */

  if (ppp_us_queue (ppp, proto, c, count+2)) {
    ppp->ddinfo.nip_rjiffies = jiffies;
    /* ERROR EN LAS FUENTES DE LINUX 1.2.13
       si entra aqui ha puesto los datos en el buffer
       y en ppp_us_queue ya ha incrementado rothers
    ppp->stats.rothers++;
    */
    return;
  }

  /* couldn't cope. */
  PRINTKN (1,(KERN_WARNING
              "ppp: dropping packet on the floor: nobody could take
it.\n"));
  slhc_toss (ppp->slcomp);
  ppp->stats.tossed++;
}

/* Examine packet at C, attempt to pass up to net layer.
   PROTO is the protocol field from the PPP frame.
   Return 1 if could handle it, 0 otherwise.  */
```

```
static int
ppp_do_ip (struct ppp *ppp, unsigned short proto, unsigned char *c,
        int count)
{
  struct device *dev;
  char nom_dev[8];
  int flags, done;

  PRINTKN (4,(KERN_DEBUG "ppp_do_ip: proto %x len %d first byte %x\n",
           (int) proto, count, c[0]));

  if (ppp_debug_netpackets) {
    PRINTK (("KERN_DEBUG %s <-- proto %x len %d\n", ppp->dev->name,
           (int) proto, count));
  }

  if ((proto == PROTO_IP) || (proto == PROTO_MLP)) {
    ppp->stats.runcomp++;
    goto sendit;
  }

  if ((proto == PROTO_VJCOMP) && !(ppp->flags & SC_REJ_COMP_TCP)) {
    /* get space for uncompressing the header */
    done = 0;
    save_flags (flags);
    cli();
    if ((ppp->rhead + 80) < ppp->rend) {
      ppp->rhead += 80;
      ppp->rcount += 80;
      done = 1;
    }
    restore_flags(flags);

    if (! done)   {
      PRINTKN (1,(KERN_NOTICE
              "ppp: no space to decompress VJ compressed TCP
header.\n"));
      ppp->stats.roverrun++;
      slhc_toss (ppp->slcomp);
      return 1;
    }

    count = slhc_uncompress(ppp->slcomp, c, count);
    if (count <= 0) {
      ppp->stats.rerrors++;
      PRINTKN (1,(KERN_NOTICE "ppp: error in VJ decompression\n"));
      slhc_toss (ppp->slcomp);
      return 1;
    }
    ppp->stats.rcomp++;
    goto sendit;
  }

  if ((proto == PROTO_VJUNCOMP) && !(ppp->flags & SC_REJ_COMP_TCP)) {
    if (slhc_remember(ppp->slcomp, c, count) <= 0) {
      ppp->stats.rerrors++;
```

```
      PRINTKN (1,(KERN_NOTICE "ppp: error in VJ memorizing\n"));
      slhc_toss (ppp->slcomp);
      return 1;
    }
    ppp->stats.runcomp++;
    goto sendit;
  }

  /* not ours */
  return 0;

 sendit:
  if (ppp_debug_netpackets) {
    struct iphdr *iph = (struct iphdr *) c;
    PRINTK ((KERN_INFO "%s <--    src %lx dst %lx len %d\n", ppp->dev-
>name,
            iph->saddr, iph->daddr, count))
  }

  /* receive the frame through the network software */
/*****************************************************
 * MODIFICADO PARA QUE PASE POR MLP SI ESTA ACTIVO *
 *****************************************************/
   sprintf(nom_dev,"mlp%ld",ppp->dev->base_addr/2);
   dev=dev_get(nom_dev);
   if ( (proto==PROTO_MLP) && ((dev->flags & IFF_UP)!=0) )
     (void) mlp_recep(c,count,0,dev);
   else
     (void) dev_rint(c,count,0,ppp->dev);
  return 1;
}


/************************************************************
 * NETWORK OUTPUT
 *    This routine accepts requests from the network layer
 *    and attempts to deliver the packets.
 *    It also includes various routines we are compelled to
 *    have to make the network layer work (arp, etc...).
 ************************************************************/

/* MODIFICADA */
int
ppp_xmit(struct sk_buff *skb, struct device *dev)
{
  struct tty_struct *tty;
  struct ppp *ppp;
  unsigned char *p;
  unsigned short proto;
  int len;

  /* just a little sanity check. */
  if (skb == NULL) {
    PRINTKN(3,(KERN_WARNING "ppp_xmit: null packet!\n"));
    return 0;
  }
```

```
  /* Get pointers to the various components */
  ppp    = &ppp_ctrl[dev->base_addr];
  tty    = ppp->tty;
  p      = (unsigned char *) (skb + 1);
  len    = skb->len;

  proto = PROTO_IP;

  PRINTKN(4,(KERN_DEBUG "ppp_xmit [%s]: skb %lX busy %d\n", dev->name,
          (unsigned long int) skb, ppp->sending));

  /* avoid race conditions when the link fails */
  if (!ppp->inuse) {
    if(skb->pkt_type==PROTO_MLP)
      return 0; /* MLP ya liberara el skb cuando sea oportuno */
    dev_kfree_skb(skb, FREE_WRITE);
    dev_close (dev);
    return 0;
  }

  if (tty == NULL) {
    PRINTKN(1,(KERN_ERR "ppp_xmit: %s not connected to a TTY!\n", dev-
>name));
    goto done;
  }

  if (!(dev->flags & IFF_UP)) {
    PRINTKN(1,(KERN_WARNING
               "ppp_xmit: packet sent on interface %s, which is down for
IP\n",
               dev->name));
    goto done;
  }

  /* get length from IP header as per Alan Cox bugfix for slip.c */
  if (len < sizeof(struct iphdr)) {
    PRINTKN(0,(KERN_ERR "ppp_xmit: given runt packet, ignoring\n"));
    goto done;
  }
  len = ntohs( ((struct iphdr *)(skb->data)) -> tot_len );

  /* If doing demand dial then divert the first frame to pppd. */
  if (ppp->flags & SC_IP_DOWN) {
    if (ppp->flags & SC_IP_FLUSH == 0) {
      if (ppp_us_queue (ppp, proto, p, len))
      ppp->flags |= SC_IP_FLUSH;
    }
    goto done;
  }

  /* Attempt to acquire send lock */
  if (ppp->sending || !ppp_lock(ppp)) {
    PRINTKN(3,(KERN_WARNING "ppp_xmit: busy\n"));
    ppp->stats.sbusy++;
    return 1;
  }
```

```
  ppp->xhead = ppp->xbuff;

/* tiene que funcionar
  if (ppp->flags & SC_COMP_TCP) {
    len = slhc_compress(ppp->slcomp, p, len, ppp->cbuff, &p,
                  !(ppp->flags & SC_NO_TCP_CCID));
    if (p[0] & SL_TYPE_COMPRESSED_TCP)
      proto = PROTO_VJCOMP;
    else {
      if (p[0] >= SL_TYPE_UNCOMPRESSED_TCP) {
      proto = PROTO_VJUNCOMP;
      p[0] = (p[0] & 0x0f) | 0x40;
      }
    }
  }
hasta aqui */

  /* increment appropriate counter */
  if (proto == PROTO_VJCOMP)
    ++ppp->stats.scomp;
  else
    ++ppp->stats.suncomp;

  if (ppp_debug_netpackets) {
    struct iphdr *iph = (struct iphdr *) (skb + 1);
    PRINTK ((KERN_DEBUG "%s ==> proto %x len %d src %x dst %x proto
%d\n",
          dev->name, (int) proto, (int) len, (int) iph->saddr,
          (int) iph->daddr, (int) iph->protocol))
  }

  /* start of frame:   FLAG  ALL_STATIONS  CONTROL  <protohi> <protolo>
*/
#ifdef OPTIMIZE_FLAG_TIME
  if (jiffies - ppp->last_xmit > OPTIMIZE_FLAG_TIME)
    *ppp->xhead++ = PPP_FLAG;
  ppp->last_xmit = jiffies;
#else
  *ppp->xhead++ = PPP_FLAG;
#endif

  ppp->fcs = PPP_FCS_INIT;
  if (!(ppp->flags & SC_COMP_AC)) {
    ppp_stuff_char(ppp, PPP_ADDRESS);
    ppp_stuff_char(ppp, PPP_CONTROL);
  }

  if(skb->pkt_type == PROTO_MLP) {
    ppp_stuff_char(ppp, PROTO_MLP&0xff);
    ppp_stuff_char(ppp, ((skb->saddr)>>8)&0xff);
    ppp_stuff_char(ppp, (skb->saddr)&0xff);
    }
  else {
    if (!(ppp->flags & SC_COMP_PROT) || (proto & 0xff00))
      ppp_stuff_char(ppp, proto>>8);
    ppp_stuff_char(ppp, proto&0xff);
    }
```

```
  /* data part */
  while (len-- > 0)
    ppp_stuff_char(ppp, *p++);

  /* fcs and flag */
  ppp_add_fcs(ppp);
  *ppp->xhead++ = PPP_FLAG;

  /* update the time for demand dial function */
  ppp->ddinfo.ip_sjiffies = jiffies;

  /* send it! */
  if (ppp_debug >= 6)
    ppp_print_buffer ("xmit buffer", ppp->xbuff, ppp->xhead - ppp->xbuff,
KERNEL_DS);
  else {
    PRINTKN (4,(KERN_DEBUG
            "ppp_write: writing %d chars\n", ppp->xhead - ppp->xbuff));
  }

  ppp_kick_tty(ppp);

 done:
  if(skb->pkt_type==PROTO_MLP)
    return 0; /* MLP ya liberara el skb cuando sea oportuno */

  dev_kfree_skb(skb, FREE_WRITE);
  return 0;
}
```

A.2. CÓDIGO ENLAZADO CON EL DAEMON PPPD

A.2.1. MAIN.C (SÓLO MODIFICACIONES)

```
/*
 * main.c - Point-to-Point Protocol main module
 */

#include "lqp.h"

extern int lqp_time;            /* TIEMPO ENTRE PAQUETES LQP */


/*
 * PPP Data Link Layer "protocol" table.
 * One entry per supported protocol.
 */
static struct protent {
    u_short protocol;
    void (*init)();
    void (*input)();
    void (*protrej)();
    int  (*printpkt)();
```

```
    char *name;
} prottbl[] = {
  { LCP, lcp_init, lcp_input, lcp_protrej, lcp_printpkt, "LCP" },
  { IPCP, ipcp_init, ipcp_input, ipcp_protrej, ipcp_printpkt, "IPCP" },
  { UPAP, upap_init, upap_input, upap_protrej, upap_printpkt, "PAP" },
  { CHAP, ChapInit, ChapInput, ChapProtocolReject, ChapPrintPkt, "CHAP" }

  { LQR, lqp_init, lqp_input, lqp_protrej, lqp_printpkt, "LQR" },
};

#define N_PROTO         (sizeof(prottbl) / sizeof(prottbl[0]))

main(argc, argv)
    int argc;
    char *argv[];
{
/* cuando termina la inicialización */

    /* MODIFICADO una vez obtenido ifunit */
    if(lqp_time >= 0)
      lqp_init(1);
    /* FIN MODIFICADO */

/* cuando se va a pasar al estado cerrado */

    if (lqp_time >= 0)
          lqp_close(0);

    quit();
}
```

A.2.2. LQP.C

```
/*
 * lqp.c - Link Quality Protocol  module
 *
 */

#ifndef lint
static char rcsid[] = "$Id: lqp.c,v 1.13 1994/05/27 01:02:14 paulus Exp
$";
#endif

#define SETSID

#include <stdio.h>
#include <string.h>
#include <signal.h>
#include <errno.h>
#include <fcntl.h>
#include <syslog.h>
#include <netdb.h>
#include <utmp.h>
#include <pwd.h>
```

```
#include <sys/wait.h>

#include <sys/ipc.h>
#include <sys/shm.h>
#include <unistd.h>

/*
 * If REQ_SYSOPTIONS is defined to 1, pppd will not run unless
 * /etc/ppp/options exists.
 */
#ifndef     REQ_SYSOPTIONS
#define REQ_SYSOPTIONS  1
#endif

#ifdef SGTTY
#include <sgtty.h>
#else
#ifndef sun
#include <sys/ioctl.h>
#endif
#include <termios.h>
#endif

#include <sys/param.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/socket.h>
#include <sys/time.h>
#include <net/if.h>

#include "callout.h"
#include "ppp.h"
#include "magic.h"
#include "fsm.h"
#include "lcp.h"

#include "pppst.h"
#include "lqp.h"

#include "ipcp.h"
#include "upap.h"
#include "chap.h"


#include "pppd.h"
#include "pathnames.h"
#include "patchlevel.h"


int lqp_time = -1 ;            /* TIEMPO ENTRE PAQUETES LQP */

__u32 OutLQRs = 0;             /* paquetes LQR transmitidos*/

__u32 InLQRs = 0;              /* paquetes LQR recibidos*/

char lqp_bufs[50];          /* estadisticas enviadas al LQP Peer*/
```

```
static struct ppp_lqp_packet *plqp_report;
                        /* estadisticas para LQP de cada enlace ppp*/
                        /* en un area de memoria compartida */

int shid;  /* identificador del segmento de memoria compartida */

char filename[40]; /* nombre del fichero shidn donde deja el shid */

extern int ifunit;

/* Inicializa LQP*/

void lqp_init (unit)
int unit;
  {
  FILE *f;

  if(unit!=1)
    return;

  if( (shid=shmget(IPC_PRIVATE, 128, IPC_CREAT | 0600)) <0 ) {
    /* realmente solo hace falta la mitad */
    perror("shmget ");
    exit(0);
    }
  sprintf(filename,"/root/mlp/shid%d",ifunit);
  if( (f=fopen(filename,"w")) == NULL) {
    perror("fopen ");
    exit(0);
    }
  fprintf(f,"%d\n",shid);
  fclose(f);

  if( (plqp_report = (struct ppp_lqp_packet*)shmat(shid, (char*)0, 0))
    == (struct ppp_lqp_packet*)(-1) ) {
      perror("shmat ");
      }

  memset ( (void *)plqp_report , 0, sizeof(struct ppp_lqp_packet));
  return;
  }

void lqp_close (unit)
int unit;
  {
  if( shmctl(shid, IPC_RMID, 0) == -1 )
    perror("shmctl ");
  unlink(filename);
  }


void lqp_input(unit,p,len)
 int unit;
 u_char *p;
 int len;

 {
```

```
   struct ppp_stats lqp_stats;

   memcpy ((void *)&(plqp_report->hdr), p+4, sizeof(struct
ppp_lqp_packet_hdr));
   if (ioctl(fd, PPPIOCGSTAT, &lqp_stats) != 0) {
    perror ("ioctl PPPIOCGSTAT");
   }
   plqp_report->tail.SaveInLQRs= ++InLQRs;
   plqp_report->tail.SaveInPackets= lqp_stats.rcomp +
               lqp_stats.runcomp + lqp_stats.rothers;
   plqp_report->tail.SaveInDiscards=lqp_stats.tossed;
   plqp_report->tail.SaveInErrors=lqp_stats.rerrors;
   plqp_report->tail.SaveInOctets=lqp_stats.rbytes;

   /* printf("Paquete LQR recibido n.%ld\n",InLQRs); */

   /*procesar*/
   if (lqp_time ==0)
      lqp_send();
   return;
}


void lqp_protrej (unit)
int unit;
{
return;
}

int lqp_printpkt(p, plen, printer, arg)
u_char *p;
int plen;
void (*printer) __ARGS((void *, char *, ...));
void *arg;
{
struct ppp_lqp_packet_hdr lqp_print;

if (plen !=48) {
  return 0;
  }

memcpy ((void *) &lqp_print, p+4, sizeof(lqp_print));
printer(arg, "%ld", 0);
printer(arg, " %ld", lqp_print.LastOutLQRs);
printer(arg, " %ld", lqp_print.LastOutPackets);
printer(arg, " %ld", lqp_print.LastOutOctets);
printer(arg, " %ld", lqp_print.PeerInLQRs);
printer(arg, " %ld", lqp_print.PeerInPackets);
printer(arg, " %ld", lqp_print.PeerInDiscards);
printer(arg, " %ld", lqp_print.PeerInErrors);
printer(arg, " %ld", lqp_print.PeerInOctets);
printer(arg, " %ld", lqp_print.PeerOutLQRs);
printer(arg, " %ld", lqp_print.PeerOutPackets);
printer(arg, " %ld", lqp_print.PeerOutOctets);
return 48;
}
```

```
/* envia un paquete de LQP*/

void lqp_send (void)
{
struct  ppp_lqp_packet_hdr  lqp_sent;
struct  ppp_stats  lqp_stats;
__u32  packets= 0;

lqp_bufs[0]= (LQR >> 8);
lqp_bufs[1]= (LQR & 0xFF);
 memset( lqp_bufs + 2, 0, 4);
memcpy(lqp_bufs +6, &(plqp_report->hdr.PeerOutLQRs), 12);
memcpy(lqp_bufs +18, &(plqp_report->tail), 20);
if (ioctl(fd, PPPIOCGSTAT, &lqp_stats) != 0) {
  perror ("ioctl PPPIOCGSTAT");
  return;
  }
OutLQRs++;
memcpy(lqp_bufs +38, &(OutLQRs), sizeof (OutLQRs));
packets= lqp_stats.scomp + lqp_stats.suncomp + lqp_stats.sothers
          +lqp_stats.serrors + lqp_stats.sbusy ;
memcpy(lqp_bufs +42, &packets, sizeof (packets));
memcpy(lqp_bufs +46, &lqp_stats.sbytes, 4);
output(0, lqp_bufs, 50);

/* printf("Paquete LQR enviado n.%ld\n",OutLQRs); */

if (lqp_time >0) {
 timeout(lqp_send, NULL, lqp_time/100);
 }
 return;
}
```

A.2.3. LQP.H

```
/*
 * lqp.h - Link Quality Protocol definitions.
 */

void lqp_init __ARGS((int));
void lqp_open __ARGS((int));
void lqp_close __ARGS((int));
void lqp_input __ARGS((int, u_char *, int));
void lqp_protrej __ARGS((int));
int  lqp_printpkt __ARGS((u_char *, int,
                  void (*) __ARGS((void *, char *, ...)), void *));

void lqp_send __ARGS((void));
```

A.3. CÓDIGO DE PROGRAMAS DE USUARIO

A.3.1. LANZAMLP.C

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/ioctl.h>
#include <net/if.h>
#include <linux/route.h>
#include <netinet/in.h>
#include <arpa/inet.h>
  /* Prototipo */
void main(void);
  /* Macros */
#define SET_SA_FAMILY(dir,fam)      \
  memset((char*)&(dir),0,sizeof(dir)); \
  dir.sa_family=(fam);

void main(void)
  {
  int s, ifunidad_mlp, activo_if_ppp[2], n, mascara_red;
  struct ifreq ifr_mlp, ifr_ppp[2];
  struct rtentry rt_mlp, rt_ppp[2];
  char mi_ip[20], su_ip[20];

  ifunidad_mlp=0;
    /* Prepara las direcciones IP y la mascara de red */
  strcpy(mi_ip,"147.156.7.107");
  strcpy(su_ip,"147.156.7.135");
  mascara_red=inet_addr("255.255.255.255");
    /* Construye nombres de los interfaces */
  sprintf(ifr_mlp.ifr_name,"mlp%d",ifunidad_mlp);
  sprintf(ifr_ppp[0].ifr_name,"ppp%d",2*ifunidad_mlp);
  sprintf(ifr_ppp[1].ifr_name,"ppp%d",2*ifunidad_mlp+1);
    /* Obtiene socket para realizar los ioctl()s */
  if ( (s=socket(AF_INET,SOCK_DGRAM,0))<0 )
    {
    printf("ERROR en socket().\n");
    exit(0);
    }
    /* Obtiene informacion sobre los interfaces */
  if ( ioctl(s,SIOCGIFFLAGS,(char*)&ifr_mlp)<0 )
    {
    printf("ERROR en ioctl(SIOCGIFFLAGS) para mlp.\n");
    exit(0);
    }
  if ( ioctl(s,SIOCGIFFLAGS,(char*)&ifr_ppp[0])<0 )
    {
    printf("ERROR en ioctl(SIOCGIFFLAGS) para ppp[0].\n");
    exit(0);
    }
  if ( ioctl(s,SIOCGIFFLAGS,(char*)&ifr_ppp[1])<0 )
```

```
    {
    printf("ERROR en ioctl(SIOCGIFFLAGS) para ppp[1].\n");
    exit(0);
    }
  for (n=0; n<2; n++)
    {
    activo_if_ppp[n]=0;
      /* Comprueba si el interface ppp esta activo */
    if ( (ifr_ppp[n].ifr_flags & IFF_UP)!=0 )
      {
      activo_if_ppp[n]=1;
      printf("El interface %s esta activo.\n",ifr_ppp[n].ifr_name);
        /* Desactiva el interface ppp */
      ifr_ppp[n].ifr_flags &= ~IFF_UP;
      if ( ioctl(s,SIOCSIFFLAGS,(char*)&ifr_ppp[n])<0 )
        {
        printf("ERROR en ioctl(SIOCSIFFLAGS) para ppp[%d]
(~IFF_UP).\n",n);
        exit(0);
        }
        /* Anula ruta al otro extremo del interface ppp */
      memset (&rt_ppp[n],'\0',sizeof(rt_ppp[n]));
      SET_SA_FAMILY(rt_ppp[n].rt_dst,AF_INET);
      SET_SA_FAMILY(rt_ppp[n].rt_gateway,AF_INET);
      rt_ppp[n].rt_dev=ifr_ppp[n].ifr_name;
      ((struct sockaddr_in*)&rt_ppp[n].rt_gateway)->sin_addr.s_addr=0;
      ((struct sockaddr_in*)&rt_ppp[n].rt_dst)-
>sin_addr.s_addr=inet_addr(su_ip);
      rt_ppp[n].rt_flags = RTF_UP | RTF_HOST;
      if ( ioctl(s,SIOCDELRT,&rt_ppp[n])<0 )
        {
        printf("ERROR en ioctl(SIOCDELRT) para ppp[%d].\n",n);
        exit(0);
        }
      }
    }
  if ( activo_if_ppp[0]!=0 || activo_if_ppp[1]!=0 )
    {
      /*******************************/
      /* Configura el interface mlp */
      /*******************************/
      /* Configura nuestra direccion IP para el interface mlp */
    SET_SA_FAMILY(ifr_mlp.ifr_addr,AF_INET);
    ((struct sockaddr_in*)&ifr_mlp.ifr_addr)-
>sin_addr.s_addr=inet_addr(mi_ip);
    if ( ioctl(s,SIOCSIFADDR,(char*)&ifr_mlp)<0 )
      {
      if ( errno==EEXIST )
        printf("ioctl(SIOSIFADDR): La direccion [%s] ya esta en
uso.\n",mi_ip);
      else
        {
        printf("ERROR en ioctl(SIOSIFADDR) para mlp.\n");
        exit(0);
        }
      }
```

```
      /* Configura la direccion IP del otro extremo para el interface mlp
*/
    SET_SA_FAMILY(ifr_mlp.ifr_dstaddr,AF_INET);
    ((struct sockaddr_in*)&ifr_mlp.ifr_dstaddr)-
>sin_addr.s_addr=inet_addr(su_ip);
    if ( ioctl(s,SIOCSIFDSTADDR,(char*)&ifr_mlp)<0 )
      {
      printf("ERROR en ioctl(SIOCSIFDSTADDR) para mlp.\n");
      exit(0);
      }
      /* Configura la mascara IP para el interface mlp */
    SET_SA_FAMILY(ifr_mlp.ifr_netmask,AF_INET);
    ((struct sockaddr_in*)&ifr_mlp.ifr_netmask)-
>sin_addr.s_addr=mascara_red;
    if ( ioctl(s,SIOCSIFNETMASK,(char*)&ifr_mlp)<0 )
      {
      printf("ERROR en ioctl(SIOCSIFNETMASK) para mlp.\n");
      exit(0);
      }
      /* Anyade la ruta al otro extremo del interface mlp */
    memset(&rt_mlp,0,sizeof(rt_mlp));
    SET_SA_FAMILY(rt_mlp.rt_dst,AF_INET);
    SET_SA_FAMILY(rt_mlp.rt_gateway,AF_INET);
    rt_mlp.rt_dev=ifr_mlp.ifr_name;
    ((struct sockaddr_in*)&rt_mlp.rt_gateway)->sin_addr.s_addr=0;
    ((struct sockaddr_in*)&rt_mlp.rt_dst)-
>sin_addr.s_addr=inet_addr(su_ip);
    rt_mlp.rt_flags = RTF_UP | RTF_HOST;
    if ( ioctl(s,SIOCADDRT,&rt_mlp)<0 )
      {
      printf("ERROR en ioctl(SIOCADDRT) para mlp.\n");
      exit(0);
      }
      /* Activa el interface mlp */
    ifr_mlp.ifr_flags = IFF_UP + IFF_POINTOPOINT + IFF_RUNNING;
    if ( ioctl(s,SIOCSIFFLAGS,(char*)&ifr_mlp)<0 )
      {
      printf("ERROR en ioctl(SIOCSIFFLAGS) para mlp (IFF_UP).\n");
      exit(0);
      }
    }
  for (n=0; n<2; n++)
    {
    if ( activo_if_ppp[n]!=0 )
      {
        /* Activa el interface ppp de nuevo */
      ifr_ppp[n].ifr_flags |= IFF_UP;
      if ( ioctl(s,SIOCSIFFLAGS,(char*)&ifr_ppp[n])<0 )
        {
        printf("ERROR en ioctl(SIOCSIFFLAGS) para ppp[%d]
(IFF_UP).\n",n);
        exit(0);
        }
      }
    }
  }
```

A.3.2. PARAMLP.C

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <net/if.h>
#include <linux/route.h>
#include <netinet/in.h>
  /* Macros */
#define SET_SA_FAMILY(dir,fam)          \
  memset((char*)&(dir),0,sizeof(dir)); \
  dir.sa_family=(fam);

main()
  {
  int s, ifunidad_mlp, activo_if_ppp[2], n, mascara_red;
  struct ifreq ifr_mlp, ifr_ppp[2];
  struct rtentry rt_mlp, rt_ppp[2];
  char mi_ip[20], su_ip[20];

  ifunidad_mlp=0;
    /* Prepara las direcciones IP y la mascara de red */
  strcpy(mi_ip,"147.156.7.107");
  strcpy(su_ip,"147.156.7.135");
  mascara_red=inet_addr("255.255.0.0");
    /* Construye nombres de los interfaces */
  sprintf(ifr_mlp.ifr_name,"mlp%d",ifunidad_mlp);
  sprintf(ifr_ppp[0].ifr_name,"ppp%d",2*ifunidad_mlp);
  sprintf(ifr_ppp[1].ifr_name,"ppp%d",2*ifunidad_mlp+1);
    /* Obtiene socket para realizar los ioctl()s */
  if ( (s=socket(AF_INET,SOCK_DGRAM,0))<0 )
    {
    printf("ERROR en socket().\n");
    exit(0);
    }
    /* Obtiene informacion sobre los interfaces */
  if ( ioctl(s,SIOCGIFFLAGS,(char*)&ifr_mlp)<0 )
    {
    printf("ERROR en ioctl(SIOCGIFFLAGS) para mlp.\n");
    exit(0);
    }
  if ( ioctl(s,SIOCGIFFLAGS,(char*)&ifr_ppp[0])<0 )
    {
    printf("ERROR en ioctl(SIOCGIFFLAGS) para ppp[0].\n");
    exit(0);
    }
  if ( ioctl(s,SIOCGIFFLAGS,(char*)&ifr_ppp[1])<0 )
    {
    printf("ERROR en ioctl(SIOCGIFFLAGS) para ppp[1].\n");
    exit(0);
    }
  for (n=0; n<2; n++)
    {
    activo_if_ppp[n]=0;
```

```
      /* Comprueba si el interface ppp esta activo */
    if ( (ifr_ppp[n].ifr_flags & IFF_UP)!=0 )
      {
      activo_if_ppp[n]=1;
      printf("El interface %s esta activo.\n",ifr_ppp[n].ifr_name);
        /* Desactiva el interface ppp */
      ifr_ppp[n].ifr_flags &= ~IFF_UP;
      if ( ioctl(s,SIOCSIFFLAGS,(char*)&ifr_ppp[n])<0 )
        {
        printf("ERROR en ioctl(SIOCSIFFLAGS) para ppp[%d]
(~IFF_UP).\n",n);
        exit(0);
        }
      }
    }
  if ( activo_if_ppp[0]!=0 || activo_if_ppp[1]!=0 )
    {
      /* Desactiva el interface mlp */
    ifr_mlp.ifr_flags &= ~IFF_UP;
    if ( ioctl(s,SIOCSIFFLAGS,(char*)&ifr_mlp)<0 )
      {
      printf("ERROR en ioctl(SIOCSIFFLAGS) para mlp (~IFF_UP).\n");
      exit(0);
      }
      /* Anula ruta al otro extremo del interface mlp */
    memset (&rt_mlp,0,sizeof(rt_mlp));
    SET_SA_FAMILY(rt_mlp.rt_dst,AF_INET);
    SET_SA_FAMILY(rt_mlp.rt_gateway,AF_INET);
    rt_mlp.rt_dev=ifr_mlp.ifr_name;
    ((struct sockaddr_in*)&rt_mlp.rt_gateway)->sin_addr.s_addr=0;
    ((struct sockaddr_in*)&rt_mlp.rt_dst)-
>sin_addr.s_addr=inet_addr(su_ip);
    rt_ppp[n].rt_flags = RTF_UP | RTF_HOST;
    if ( ioctl(s,SIOCDELRT,&rt_ppp[n])<0 )
      {
      printf("ERROR en ioctl(SIOCDELRT) para mlp.\n");
      exit(0);
      }
      /*********************************************/
      /* Anula las direcciones IP del interface mlp */
      /*********************************************/
    SET_SA_FAMILY(ifr_mlp.ifr_addr,AF_INET);
    SET_SA_FAMILY(ifr_mlp.ifr_dstaddr,AF_INET);
    SET_SA_FAMILY(ifr_mlp.ifr_netmask,AF_INET);
      /* Configura nuestra direccion IP para el interface mlp */
    ((struct sockaddr_in*)&ifr_mlp.ifr_addr)-
>sin_addr.s_addr=inet_addr("0.0.0.0");
    if ( ioctl(s,SIOCSIFADDR,(char*)&ifr_mlp)<0 )
      {
      if ( errno==EEXIST )
        printf("ioctl(SIOSIFADDR): La direccion 0.0.0.0 ya esta en
uso.\n");
      else
        {
        printf("ERROR en ioctl(SIOSIFADDR) para mlp.\n",n);
        exit(0);
        }
```

```
      }
      /* Configura la direccion IP del otro extremo para el interface mlp
*/
    ((struct sockaddr_in*)&ifr_mlp.ifr_dstaddr)-
>sin_addr.s_addr=inet_addr("0.0.0.0");
    if ( ioctl(s,SIOCSIFDSTADDR,(char*)&ifr_mlp)<0 )
      {
      printf("ERROR en ioctl(SIOCSIFDSTADDR) para mlp.\n",n);
      exit(0);
      }
      /* Configura la mascara IP para el interface mlp */
    ((struct sockaddr_in*)&ifr_mlp.ifr_netmask)-
>sin_addr.s_addr=inet_addr("0.0.0.0");
    if ( ioctl(s,SIOCSIFNETMASK,(char*)&ifr_mlp)<0 )
      {
      printf("ERROR en ioctl(SIOCSIFNETMASK) para mlp.\n",n);
      exit(0);
      }
    }
  for (n=0; n<2; n++)
    {
    if ( activo_if_ppp[n]!=0 )
      {
        /* Anyade la ruta al otro extremo del interface ppp[n] */
      memset(&rt_ppp[n],0,sizeof(rt_ppp[n]));
      SET_SA_FAMILY(rt_ppp[n].rt_dst,AF_INET);
      SET_SA_FAMILY(rt_ppp[n].rt_gateway,AF_INET);
      rt_ppp[n].rt_dev=ifr_ppp[n].ifr_name;
      ((struct sockaddr_in*)&rt_ppp[n].rt_gateway)->sin_addr.s_addr=0;
      ((struct sockaddr_in*)&rt_ppp[n].rt_dst)-
>sin_addr.s_addr=inet_addr(su_ip);
      rt_ppp[n].rt_flags = RTF_UP | RTF_HOST;
      if ( ioctl(s,SIOCADDRT,&rt_ppp[n])<0 )
        {
        printf("ERROR en ioctl(SIOCADDRT) para ppp[%s].\n",n);
        exit(0);
        }
        /* Activa el interface ppp */
      ifr_ppp[n].ifr_flags |= IFF_UP;
      if ( ioctl(s,SIOCSIFFLAGS,(char*)&ifr_ppp[n])<0 )
        {
        printf("ERROR en ioctl(SIOCSIFFLAGS) para ppp[%d]
(IFF_UP).\n",n);
        exit(0);
        }
      }
    }
  }
```

A.3.3. MONITOR.C

```
#include <stdio.h>
#include <sys/ipc.h>
#include <sys/shm.h>
```

```
#include <signal.h>
#include <sys/time.h>

#include "pppst.h"

#define NUM 2

struct ppp_lqp_packet *plqp[2];
int acti[NUM];

void mnj_int(void) {
  printf("SIGINT\n");
  exit(0);
  }

void alrm(void) {
int i;

  for(i=0;i<NUM;i++)
    if(acti[i]) {
      printf("ppp%d      %10s  %10s  %10s  %10s\n\n",i,
        "LastOut","PeerIn","PeerOut","SaveIn");
      printf( "LQRs      %10ld  %10ld  %10ld  %10ld\n",
         plqp[i]->hdr.LastOutLQRs, plqp[i]->hdr.PeerInLQRs,
         plqp[i]->hdr.PeerOutLQRs, plqp[i]->tail.SaveInLQRs);
      printf( "Packets   %10ld  %10ld  %10ld  %10ld\n",
         plqp[i]->hdr.LastOutPackets, plqp[i]->hdr.PeerInPackets,
         plqp[i]->hdr.PeerOutPackets, plqp[i]->tail.SaveInPackets);
      printf( "Octets    %10ld  %10ld  %10ld  %10ld\n",
         plqp[i]->hdr.LastOutOctets, plqp[i]->hdr.PeerInOctets,
         plqp[i]->hdr.PeerOutOctets, plqp[i]->tail.SaveInOctets);
      printf( "Discards  %10s  %10ld  %10s  %10ld\n",
         "", plqp[i]->hdr.PeerInDiscards,
         "", plqp[i]->tail.SaveInDiscards);
      printf( "Errors    %10s  %10ld  %10s  %10ld\n\n",
         "", plqp[i]->hdr.PeerInErrors,
         "", plqp[i]->tail.SaveInErrors);
      }
    else
      printf("no es posible la monitorizacion de ppp%d\n", i);
  printf("\n\n\n\n\n\n\n");
  signal(SIGALRM, alrm);
  }

main(void)
  {
  FILE *f;
  char filename[20];
  int shid, i, cuantos;
  struct itimerval itv;

  signal(SIGINT, mnj_int);
  signal(SIGALRM, alrm);

  for(i=0;i<NUM;i++) {
    acti[i]=0;
    sprintf(filename,"/root/mlp/shid%d",i);
```

```
    if( (f=fopen(filename,"r")) == NULL) {
      perror("fopen ");
      continue;
      }
    if(fscanf(f,"%d",&shid)!=1) {
      fprintf(stderr,"no pudo leer el shid%d\n",i);
      fclose(f);
      continue;
      }
    fclose(f);

    if( (plqp[i] = (struct ppp_lqp_packet*)shmat(shid, (char*)0, 0))
         == (struct ppp_lqp_packet*)(-1) ) {
      perror("shmat ");
      continue;
      }
    acti[i]=1;
    }

  cuantos=0;
  for(i=0;i<NUM;i++)
    if(acti[i])
      cuantos++;

  if(cuantos==0) {
    fprintf(stderr,"No hay ningun comunicante activo\n");
    exit(0);
    }

  itv.it_interval.tv_sec = 1;
  itv.it_interval.tv_usec = 0;

  itv.it_value.tv_sec = 1;
  itv.it_value.tv_usec = 0;

  if (setitimer(ITIMER_REAL, &itv, NULL)) {
    perror("setitimer(ITIMER_REAL):");
    }

  while(1)
    pause();
  }
```

A.3.4. TOLM.C

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/ioctl.h>
#include <net/if.h>
#include <linux/route.h>
#include <netinet/in.h>
```

```
#include <arpa/inet.h>

#include <linux/mlp.h>

  /* Macros */
#define SET_SA_FAMILY(dir,fam)      \
  memset((char*)&(dir),0,sizeof(dir)); \
  dir.sa_family=(fam);

void main(argc,argv)
int argc;
char **argv;
  {
  int s, modo, modo0;
  struct ifreq ifrm;

  sprintf(ifrm.ifr_name,"mlp0");

  /* Obtiene socket para realizar los ioctl()s */
  if ( (s=socket(AF_INET,SOCK_DGRAM,0))<0 )
    {
    printf("ERROR en socket().\n");
    exit(0);
    }

  /* Obtiene informacion sobre los interfaces */
  if ( ioctl(s,SIOCGIFFLAGS,(char*)&ifrm)<0 )
    {
    printf("ERROR en ioctl(SIOCGIFFLAGS) para mlp.\n");
    exit(0);
    }

  if(ifrm.ifr_flags & IFF_UP)
    printf("El interfaz %s esta activo.\n",ifrm.ifr_name);
  else
    printf("El interfaz %s esta inactivo.\n",ifrm.ifr_name);

  if ( ioctl(s,SIOCGTOLM,(char*)&ifrm)<0 )
    {
    printf("ERROR en ioctl(SIOCGTOLM) para %s.\n",ifrm.ifr_name);
    exit(0);
    }
  modo0 = ifrm.ifr_metric;

  if( (argc<2) || ((modo=atoi(argv[1]))==0) ) {
    printf("Modo del interfaz %s: %d\n",ifrm.ifr_name,modo0);
    exit(0);
    }

  ifrm.ifr_metric = modo;
  if ( ioctl(s,SIOCSTOLM,(char*)&ifrm)<0 )
    {
    printf("ERROR en ioctl(SIOCSTOLM) para %s.\n",ifrm.ifr_name);
    exit(0);
    }

  if ( ioctl(s,SIOCGTOLM,(char*)&ifrm)<0 )
```

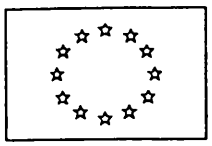
```
    {
    printf("ERROR en ioctl(SIOCGTOLM) para %s.\n",ifrm.ifr_name);
    exit(0);
    }
  modo = ifrm.ifr_metric;

  printf("Modo del interfaz %s: %d -> %d \n",ifrm.ifr_name,modo0,modo);

  /*
  ifrm.ifr_flags = IFF_UP + IFF_POINTOPOINT + IFF_RUNNING;
  if ( ioctl(s,SIOCSIFFLAGS,(char*)&ifrm)<0 ) {
    printf("ERROR en ioctl(SIOCSIFFLAGS) para mlp0 (IFF_UP).\n");
    exit(0);
    }
  */

  }
```

A.3.5. BERM.C

```
/*
 programa que introduce errores en un interfaz ppp
 Uso: berm interfaz(pppn) velocidad ber(0.00005-0.000001)
*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/ioctl.h>
#include <net/if.h>
#include <linux/route.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <signal.h>
#include <sys/time.h>

struct ifreq ifrm;
double ber = 0;
int speed, s;

static void
alrm(sig)
int sig;
  {
  struct itimerval itv;
  double t;  /*tiempo entre errores*/

#ifdef DEBUG
  printf("ALARMA\n");
#endif

  ifrm.ifr_metric = 256;
```

```
  if ( ioctl(s,SIOCSIFMETRIC,(char*)&ifrm)<0 )
    {
    printf("ERROR en ioctl(SIOCSIFMETRIC) para %s.\n",ifrm.ifr_name);
    exit(0);
    }

  t=(rand()*2.0)/(speed*ber*RAND_MAX); /* aleatorio de media
1/(ber*speed) */

#ifdef DEBUG
  printf("Proxima alarma en %.6fs\n",t);
#endif

    itv.it_interval.tv_sec = 0;
    itv.it_interval.tv_usec = 0;

    itv.it_value.tv_sec = (long)t;
    itv.it_value.tv_usec = (long)(1000000.0*(t-(long)t));

    if (setitimer(ITIMER_REAL, &itv, NULL)) {
      perror("setitimer(ITIMER_REAL):");
     }

    signal(SIGALRM,alrm);
}


void main(argc,argv)
int argc;
char **argv;
  {
  struct itimerval itv;
  double t;  /*tiempo entre errores*/

  if( (argc != 4) || (strncmp(argv[1],"ppp",3)) ||
((speed=atoi(argv[2]))==0)
    || ((ber=atof(argv[3]))<0.000001) || (ber>0.00005) ){
    printf("ERROR. Uso: berm interfaz(pppn)  velocidad ber\n");
    exit(0);
    }

  strcpy(ifrm.ifr_name, argv[1]);

  /* Obtiene socket para realizar los ioctl()s */
  if ( (s=socket(AF_INET,SOCK_DGRAM,0))<0 )
    {
    printf("ERROR en socket().\n");
    exit(0);
    }

  /* Obtiene informacion sobre el interfaz */
  if ( ioctl(s,SIOCGIFFLAGS,(char*)&ifrm)<0 )
    {
    printf("ERROR en ioctl(SIOCGIFFLAGS) para %s.\n",ifrm.ifr_name);
    exit(0);
    }
```

```
  if(ifrm.ifr_flags & IFF_UP)
    printf("\nEl interfaz %s esta activo.\n",ifrm.ifr_name);
  else
    printf("\nEl interfaz %s esta inactivo.\n",ifrm.ifr_name);

  if ( ioctl(s,SIOCGIFMETRIC,(char*)&ifrm)<0 )
    {
    printf("ERROR en ioctl(SIOCGIFMETRIC) para %s.\n",ifrm.ifr_name);
    exit(0);
    }

  printf("Modo del interfaz %s: %d\n",ifrm.ifr_name, ifrm.ifr_metric);

  signal(SIGALRM,alrm);

/*
      sigemptyset(&mask);
      sigaddset(&mask, SIGALRM);

      sa.sa_mask = mask;
      sa.sa_flags = 0;
      sa.sa_handler = alrm;
      if(sigaction(SIGALRM,&sa,NULL) < 0)
        perror("sigaction:");
*/

      t=(rand()*2.0)/(speed*ber*RAND_MAX); /*aleatorio de media
1/(ber*speed)*/

#ifdef DEBUG
  printf("Proxima alarma en %.6fs\n",t);
#endif

      itv.it_interval.tv_sec = 0;
      itv.it_interval.tv_usec = 0;

      itv.it_value.tv_sec = (long)t;
      itv.it_value.tv_usec = (long)(1000000.0*(t-(long)t));

      if (setitimer(ITIMER_REAL, &itv, NULL)) {
        perror("setitimer(ITIMER_REAL):");
        }

      while(1)
        pause();
}
```



EUROPEAN COMMISSION
Directorate-General for Transport
Directorate-General for Telecommunications,
Information Market and Exploitation of Research

ANEXO B

RTT-HLG
HLG5/EC/BD35

ROAD TRANSPORT TELEMATICS

HIGH-LEVEL GROUP

Review of US National ITS Architecture

Draft 7 February 1997

Review of US National ITS Architecture

Executive Summary

Draft 7/2/97

Report produced by a team of experts convened by the European Commission (DG XIII):

Rapporteur: Mr. Lång (Ericsson)

Mr. Bonora (Marconi)
Mr. Vis (Rijkswaterstaat)
Mr. Finn (ETT)
Mr. Bossom (Siemens)
Mr. Fisher (GF Consultancy)
Mr. Pagny (DSCR-METT)
Mr. Brand (Brand Consult.)
Mr. Rennesson (AFT-TNT)
Mr. Munck (Volvo)
Mr. Glathe
Mr. Kossack (Siemens)
Mr. Pfliegl (Alcatel)
Mr. von Pattay (ERTICO)

NOTICE

This document results from the review of the US National ITS Architecture carried out by a team of experts convened by the European Commission (DG XIII). It reflects as well the discussions held between this group of experts, other invited experts and European Commission officials.

The opinions expressed in this document do not necessarily represent the views of the European Commission services.

TABLE OF CONTENTS

Forth Draft Report

Review of US National ITS Architecture

1 Introduction

Development of Intelligent Transport Systems (ITS) in America was given a tremendous boost in 1991 with the Intermodal Surface Transport Efficiency Act, the law that formalised the American ITS Programme. Amongst other things it stimulated the production of a national plan for improving surface transportation. One part of this plan was a study programme which resulted in the development of a US National ITS Architecture for transportation systems. The result of this work is the subject of this report. **This $25,000,000 programme started in October 1993 and was completed in June 1996.** The resulting Architecture has been analysed and its consequences for Europe are addressed in this report.

2 Architecture Background and Development

So what is the Us National ITS Architecture about? **It provides a common structure for the design of intelligent transport systems.** It is not intended to be a design but to be a framework around which multiple design approaches can be developed, each one tailored to meet individual needs for the user. The Architecture defines the functions (e.g. gather traffic information) that must be performed to implement a given user service (e.g. traffic control), the physical entities or subsystems where these functions reside (e.g. the roadside or the vehicle), the information flows between the physical subsystems, and the communication requirements for the information flows. In addition, it identifies and specifies the requirements for the interface standards needed to support national and regional interoperability, as well as product standards needed to support economy of scale considerations for equipment supply [1]

Phase 1 of the system architecture study (1993-1994), which was completed in October 1994, involved four consortia led by the Hughes Aircraft Company, Loral Federal Systems, Rockwell International and Westinghouse Electric respectively. Each independently developed their concept of a US National ITS Architecture based on the Federal Highways Administration (FHWA) specification for the 29 User Service Requirements (USR's) - see Annex 1). All four consortia analysed the USR's and their inter-dependencies using established systems analysis techniques. They then developed a coherent vision of the functions to fulfil the USR's might be developed over time, implemented and operated. After evaluation of Phase 1 results at the end of 1994, two consortia - Loral and Rockwell - were retained for Phase II of the programme. This started in February 1995 and the consortia were asked to continue development of their architectures and incorporate features from those developed by the other two consortia. The Loral and Rockwell consortia decided from the outset that the most sensible course of action was to combine their efforts into the Architecture Development Team. This produced the final version of the architecture in June 1996. It was developed from the best parts of those provided by all four of the Phase I consortia and included new features and facilities to

[1] ITS Architecture - Executive summary, June 1996

improve its acceptability and compliance with revised User Service Requirements produced in July 1994 by the FHWA.

During this whole Architecture definition process, an important activity was launched across the USA to reach out and gain stakeholders' inputs and comments. There have been two major system architecture workshops: one to define the relationship of the system architecture to the volunteer standard-setting process used in the US and the other to engage all ITS America committees in detailed architecture discussions and feedback to the architecture development consortia. In addition, architecture fora were organised and over 25 outreach meetings were held across the US producing comments that have been incorporated into the programme.

The final result of this consolidation has been published in June 1996. It is composed of 5574 pages in 18 documents. However the FHWA are ensuring that the Architecture does not remain static. They have funded further work to enhance the Architecture through the introduction of an additional User Service Requirement for the Highway Rail Interface (HRI). The current Architecture is available on the Internet http://www.rockwell.com/itsarch/.

In addition to the HRI work, the FHWA will fund Phase III of the US National ITS Architecture programme. This will be designed to maintain and develop the Architecture, feeding in any changes produced as a result of the standards setting programme and other ITS activities.

3 Purpose of this review

Following the production of the US National ITS Architecture, there was a clear need for Europe to understand the possible consequences of its appearance and to initiate actions, if needed. These actions must also bear in mind that ITS world markets are growing and are likely to be influenced by the US National ITS Architecture as will be the ISO standardisation process.

In August 1996, the European Commission (DG XIII) launched an activity to review the US National ITS Architecture, to compare it to European achievements in Advanced Transport Telematics. It would also define a common European position that could be discussed in a later stage with the main European stakeholders.

The aim of the review was to provide answers to the following questions:

- What are the strengths and weaknesses of the US National ITS Architecture compared to the European results?
- What are the implications of the ITS Architecture results for Europe?
- Are there aspects of the US National Architecture that Europe should welcome and support? Are there other aspects which Europe should resist, or seek to modify?
- Are there any European results that would improve the US National ITS Architecture and could contribute to a global approach?
- Is it possible to identify stakeholders who could be impacted by the American results?
- What could be an appropriate European response in the immediate and short term (1-2 years) and medium term (> 2 years)?
- What could be recommended for further European and international work? What actions are needed in research programmes and/or at the standardisation level?

The present document summarises the key findings of the review team based on the topics aforementioned.

/dokument/usarch/usrev03.doc

4 Approach followed for the review

A team of experts has been identified by the European Commission. These experts received guidelines from the European Commission for their work and a complete set of US National ITS Architecture documents. These experts covered the following areas of expertise:

Automatic Debiting System (ADS)	Mr. Bonora (Marconi) Mr. Vis (Rijkswaterstaat)
Public Transport (PT)	Mr. Finn (ETT)
Urban Transport (UT)	Mr. Bossom (Siemens) Mr. Lång (Ericsson)
Inter-Urban Transport (IUT)	Mr. Fisher (GF Consultancy) Mr. Pagny (DSCR-METT)
Freight and Fleet Management (FFM)	Mr. Brand (Brand Consult.) Mr. Rennesson (AFT-TNT)
Vehicle Control (VC)	Mr. Munck (Volvo) Mr. Glathe
Information and Communication Systems (IS/CS)	Mr. Kossack (Siemens) Mr. Pfliegl (Alcatel)
Contribution for Standardisation	Mr. Kossack (Siemens) Mr. von Pattay (ERTICO)

A first meeting was organised by the European Commision in September 1996 to allow the experts to have a briefing on the activity and to share problems related to it.

The experts individually reviewed the US National ITS Architecture documentation and wrote reports with their findings related to the areas that they had been allocated, under the headings described in the previous section of this report.

The European Commission organised a workshop in October 96. First results from the reviews produced by the team of experts were presented to and discussed with European Commission officials and invited experts.

The present report has been compiled based on the consolidated results of all the reviews.

The next phase will include the organisation of events during which the results of this work will be presented to the main stakeholders' representatives. The document will also be submitted to the High Level Group on Road Transport Telematics of representatives from the Member States.

5 Key findings

5.1 Key features of US National ITS Architecture

General remarks:

The US National ITS Architecture can be viewed as an instrument to get the ITS market in the US moving with the speed and focus necessary to gain the critical mass for ITS in the US first. The political message "now we start" is as important as the technical content of the architecture. It has been launched by the Federal Government and has a comprehensive coverage of user services. The Architecture proves a systematic approach to the process, it is

complete, consistent and covers a broad range of analysis and views. The homogenous appeal of the whole Architecture presentation is a tremendous strength from the marketing point of view (export) as compared to the EU where a multitude systems and solutions may make potential customers unsure.

The US National ITS Architecture covers a time horizon of about 16 years from now. It takes into consideration the interdependencies between transportation and society, and the needs to develop an adequate telecommunication system. It addresses road transport in most forms and provides links to other modes.

US National ITS Architecture documentation is good, it is available to everybody and it is valuable to read (but difficult to navigate without prior knowledge). It presents Architecture features and functions in a way that can be understood by those without expertise in transport.

The US ITS Architecture provides a framework from which products can be developed without specifying the technology that is to be used. It is open to be implemented by different designs so long as they fulfil the specified functionality. In addition it has a very clear view of how industry should deploy the resulting products with customers. Market Packages have been proposed to create an environment for an ITS market.

There is nothing about the "look and feel" of the human-machine interfaces, as they were considered to be part of the detailed design. However the HMI design is critical for user acceptance, one of the main areas of competition for industry.

The US National ITS Architecture does not address organisation issues because they were considered outside the scope of the architecture and are too complex. But Institutional hurdes are identified, highlighted and then passed on as potential problems to responsible people. The US national ITS Architecture stresses the concept of "Information Service Provider", an organisation providing travellers with a variety of services. This organisation may be part of a jurisdictional authority, or a private company.

Area specific remarks

For Automatic Debiting Systems:

- Integrated Payment System (IPS) is not completely developed (e.g. the usage of electronic payment as a tool for demand management, in public transport and the aspects of intermodality)
- It emphasises the concept of advance payment and not the concept of chained fare product. Payment instruments and financial institutions are considered as terminators outside the architecture.
- Adoption of the draft CEN 5.8 GHz standard is possible (DSRC move up to the 5.8 GHz band is mentioned).

For Public Transport:

- Fare collection is biased to electronic payment.
- Demand responsive systems are only partly covered in the sense of advising the ride requirement, but it does not appear to extend back into the practicalities of procuring

For Urban Transport:

- A wide variety of traffic management strategies are supported, including those for parking management and control of vehicle induced pollution.
- Road pricing through tolling is included, but given low priority.

- The management of incidents is given great priority in order that their impact on the whole transportation network may be minimised.
- The use of smart cards and driver licences for access control is not included in the User Services.

For Inter-Urban Transport:

- Strategies for the management and control of inter-urban traffic are provided.
- The main area of attention for the Inter-Urban network relates to the use of wireless communications in almost all applications and services.
- The use of probe data bas been adopted, for the collection of network information data. However its application does not appear to have been clearly researched. This is a change of concept for "traditional" traffic engineers who have built their systems on the concept of collecting data from fixed point traffic count stations.

Freight and Fleet Management:

- It focuses on enhancing truck mobility and reducing the time that they are stopped for inspection, i.e. interaction between freight transport industry and administrations.
- Commercial and operational freight and fleet management are provided to a limited extent.
- CVO (Commercial Vehicle Operations) infrastructure only based on DSRC (Dedicated Short Range Communications) along the highway.

Vehicle Control:

- European work on architecture and system integration is lagging behind the US. There has been however a substantial amount of technical development has been carried out in Europe.
- Co-operative driving is properly addressed. For safety reason the US National ITS Architecture assumes a combination of road side and in-vehicle intelligence. How to achieve an optimum depends on the results of large scale tests. The Architecture leaves space for different designs.
- Advanced vehicle safety systems not fully addressed. Implementation decisions in the physical architecture are left to the manufacturers.

Communication and information systems:

- American approach cannot be applied to the European scene on a single country basis due to the complexity of the various national situations. A co-ordinated European approach is necessary.
- European results are closer to its market but the lack of a common vision is a handicap for Europe.
- Its approach to communications, which separate the transportation layer from the application layer conforms with OSI.

Standardisation:

- The study is sceptical about the desirability and feasibility of international standards for all service definitions, interfaces and protocols needed in ITS, due to legal, regulatory and cultural differences between countries.
- Development of the US market and US standards are given a high priority in the document[2]

5.2 Strengths and weaknesses of the US approach

Strengths:

- It has powerful support from Federal Government and other organisations such as the AASHTO.
- It provides one single and consistent overview of the framework for the development of ITS. It is available to use and easy accessible.
- It provides planning confidence to the great diversity of players in the US market place. The Architecture provides enough functional details for the next step of product design to be started. The support of the Government make the players confident the Architecture will be reliable and stable.
- It is a well founded document for further discussion and development.

Weaknesses

- Some of the user services lack maturity. They give no real idea of what is required.
- Compatibility with the US National ITS architecture does not yet guarantee the compatibility of products/interoperability of services.
- The communication architecture document covers more or less all state-of-the-art technologies and does not favour, for cost or performance reasons, a limited range of technologies to be recommended for further developments.

5.3 Suitability and relevance of the US National ITS Architecture for Europe

Most of the US National ITS Architecture may be used in Europe but the User Needs and the focus will need to be different. User services and user needs is the right starting point for European architecture development. Once the European architecture has been developed, the next step will be to shift the focus to standardisation, particularly of communications interfaces.

The following issues are different in Europe and are therefore not addressed in the US National ITS Architecture:

- The institutional framework is different in Europe.
- The language problem is not addressed.

---

[2] The NTCIP protocol is specified to be used for the transfer of data between the roadside and operating centres and between operating centres themselves. This protocol is a good candidate for an international standard.

- There is not enough focus on public transport and intermodality.
- Integrated payment is more central for automatic debiting systems in EU, but interoperability between on-board unit and roadside as a final goal is the same.
- Very few European standards are referenced in the Standards Catalogue. 14 % of the listed specifications are international standards and 1,5 % national standards from outside the US.
- RDS/TMC has been given a low priority.
- In the US Freight and Fleet Management (FFM) is focused on reducing the time trucks are stopped for inspection. The work in Europe has put emphasis on the reduction of the amount of empty kilometres in road freight, and the development of intermodality. In the US FFM systems including on board equipment are in place, interfaces to public administrations are under way and the interest in intermodal FFM applications has just started. In Europe the situation is different: Company FFM systems are under implementation, interfaces to administrations are not being developed and the intermodal applications are already underway.
- The user needs in urban transport differ between US and EU, due to difference in city planning and population density. There is a more widespread knowledge on transport management and advanced technology in European cities.
- Significant differences are observed between the role of standards in the US and Europe. Procurement in Europe is guided by standards including "voluntary" standards to much higher degree than the US market.

We should include in the European architecture development, the European strengths such as a wide spread knowledge on transport management and advanced technology in cities and interurban traffic control. It is also a European strength to have communication systems with pan European coverage (i.e. RDS, ISDN and GSM). It is the developers who bring an architecture into effect. That has to be seen if the US top-down approach with a clear vision is more effective than the European "simultaneous" approach. However the simultaneous approach must be led by a goal to be efficient.

5.4 Relevant European results

There is no consensus on European System Architecture results (e.g. SATIN, TELTEN) although many results are already available, see annex 2.

The lack of common architecture vision in Europe can be explained if we look at the genesis of the SATIN task force. This was called upon to build on previously available results from European projects. The US chose to adopt "greenfield" approach coupled with an implementation time frame of 20 years (end date is 2012). This provided complete freedom for the architecture to specify new developments and time in which to implement a migration path from the current diverse architectures.

The horizontal project CONVERGE in the Telematic Application Programme in the fourth framework programme is helping projects define, develop and harmonise the documentation and presentation of their architectures. This approach will not by itself alone provide, in a systematic way, the interoperability of services or compatibility of products in Europe.

The initial CEN TC 278 workplan (1991) was the first attempt to establish a top-down systems architecture and it has lead to a number of consensus documents and pre-ENVs.

5.5 Stakeholders

The main stakeholders identified across all areas are the public transport authorities, the public road authorities, the motor vehicle industry, the transport equipment industry, the telecommunications industry and the multi-national IT-industry.

5.6 Problem areas for Europe

Europe lacks (but the US has) a clear vision of why it needs to have a single transport telematics architecture. Without this vision it is pointless going ahead with any further work on the development of an architecture.

There are too many views in Europe on what is an architecture, what it should cover and what it should be used for. Local actors and industries are afraid of a Systems Architecture, because it is perceived that it may stop and delay their plans and (for industry) erode hard won market.

Europe is missing a powerful promoter/developer of the systems architecture. It has not been seen as a top priority issue in Europe despite the inclusion of the topic in the Call for Proposals of the 3rd and 4th Framework Programmes as shared cost activities. This is one of the reasons why the problem outlined in the previous two paragraphs exist.

Nothing like ITS America and FHWA (Federal Highway Administration) exists in Europe to push systems architecture. The High Level Group established by the Commission could act as a promoter but input from private sector must be secured.

In Europe, the understanding that "simultaneous engineering" between public agencies, private companies and user associations does not interfere with the responsibilities each one of them has in his particular sector is not properly understood and developed. Enabling these groups to participate will give them a feeling of ownership, that can only help increase the acceptance of an architecture across Europe.

Any European architecture will have to address the complex problems of data protection and privacy. This issue is further complicated by the legal constraints that apply in these areas. The architecture must be flexible in the way it classifies data and uses it for enforcement purposes.

The architecture must allow for an alternative to sending identifying data to enforcement agencies in violation handling due to legal constraints. The flexibility in classification and enforcement must be built into the architecture.

The architecture must be capable of handling truly intermodal transport solutions as understood within European transport policy.

6 Proposed European actions

The reviewers propose a European approach based on the following points:

- Europe must develop and promote a framework for products and services This could be called the European Transport Telematics Architecture. It would be a top down guidance to achieve fully integrated systems. There is no need for a European ITS Architecture competing with the US ITS Architecture. Instead the general European approach should aim for to include as much as possible of the US architecture. However, the European Transport Telematics Architecture should be based on European User needs, include European results and take account of what already exists in Europe. A completely new

exercise is not necessary but there should be a collection and adaptation of existing documentation into a consistent document.

- A co-ordinated European action is needed. It will not be possible for Europeans to gain the critical mass for one of their specifications before US does, unless the Europeans co-operate, agree on common specifications and implement them without undue delay. A powerful promoter/developer should initiate the development of a European business plan for Transport Telematics including a plan for deployment and a long term strategy for standardisation. The promoter/developer must also make his commitment visible (the US Transport Minister Peña's initiative "Operation Time Saver" is a good example).
- CEC DGXIII should influence more the CEN/TC 278 work. Specifically WG13 should put more effort in co-ordinating the other Working Groups and try to separate application level from communication level in order to make standards more technology independent. The Commission should also support CEN TC 278 WG 13 to promote a European view on architecture in ISO.
- Reports produced in European public funded R&D projects should be made widely available using means such as the Internet.

Proposed immediate European actions:

- Provide a powerful promoter/developer for the whole programme to develop and deploy a European Transport Telematics Architecture. This body could be funded as part of the EC R&D programme and be responsible for organising the completion of the following activities:

  - production of a set of User Needs for Europe which are to be satisfied by the Architecture and which include target date (or dates) by which they must be satisfied;
  - the actual development of the European Transport Telematics Architecture, including an analysis of communications needs, costs, benefits, risks, deployment plans, etc.;
  - the activation of the standards creation activities for interfaces defined by the Architecture;
  - promotion of outreach and consensus forming activities to ensure that Europe will take ownership of the User Needs, the Architecture, resulting standards, and actively promote their deployment and implementation;
  - generation, together with key stakeholders, of a business plan for Transport Telematics products and services;
  - establishment of a time scale for the above activities, highlighting their various completion dates, and taking into account the target date(s) specified for the satisfaction of the User Needs;
  - promote the topic with EU Member States' governments to ensure that they are all actively involved and in favour of the activities outlined above;
  - the recruitment of individuals and/or organisations to assist and participate in the above activities, subject to funding from and the approval of the EC.

- There must be a clear decision point for the course of action to be adopted by Europe, and an accepted implementation Plan with the needed resources allocated to it. Key players from the different sectors should draft a business plan for Transport Telematics, under the lead of a promoter/developer.
- The Commission is requested to improve the running of horizontal actions in DGVII and DG XIII and others with regard to Transport Telematics in Europe in order to ensure the achievement of a co-ordinated strategy.
- Support CEN TC 278 WG 13 to promote a European view on architecture in ISO.

Proposed European actions in the short term (1-2 year)

- Define a European Transport Telematics Architecture based on US and European work.
- Support European standardisation in CEN. The public standards are developed in a democratic process and are based on consensus of all interested parties. This is a strength but the resulting standards development process takes too long to implement. One of the reasons is that additional financial resources are needed as standardisation is mostly driven by volunteers, setting aside days from their already busy schedules to do unpaid-for work. A number of improvement of the standardisation process have to be made in order to speed it up (see annex 11).
- Form a concertation framework by actively involving stakeholders such as infrastructure owners, cities using Transport Telematics today (i.e. the POLIS network) and the new Information Service Providers (ISP's) in the Architecture development process.
- Promote transport and traffic information services via GSM as additional pan-European service to RDS/TMC. The pan-European services could provide important feed back from deployment of designs included in the Architecture.
- Develop and foster skills and training in the domain of system architecture to support European industry.
- Carry out market characteristics and market opportunity studies in USA and Pacific Rim countries to assist European industry to increase its penetration of the world-wide ITS market.
- The European Commission should enhance the rules for public procurement. This is to enable public procurement to overcome the effects of purchasing products and systems with proprietary interfaces. The change to the rules would require the supplier to commit himself to provide the specification of vital interfaces and protocols for public standardisation or to license everybody at fair and non-discriminatory conditions for any patent needed to supply subsystem to inter-work.

The actions in some of the bulleted items above will form part of the work for the promoter/developer role.

Proposed European actions in the long term (>2 year)

- Maintain and develop the architecture based on real designs and new user needs including further consideration of intermodal aspects.
- Develop and promote standards for European road pricing as a means to further increase the pan-European key Transport Telematic services.

The actions in some of the bulleted items above will form part of the work for the promoter/developer role.

Annexes

The following annexes have been produced as part of this report. They are not included in the draft Executive Summary distributed to the High Level Group on Road Transport Telematics and will be provided at a later date.

1. User Services
2. An overview of Programme level achievements on Systems Architecture, CONVERGE project.
3. The need for a European Transport Telematics Architecture. Benefits and Opportunities. CONVERGE project.
4. Area Automatic Debiting System (ADS) review report, Bonora (Marconi), Vis (Rijkswaterstaat)
5. Area Public Transport (PT) review report, Finn (ETT)
6. Area Urban Transport (UT) review report, Bossom (Siemens), Lång (Ericsson)
7. Area Inter-Urban Transport (IUT) review report, Fisher (GF Consultancy), Pagny (DSCR-METT)
8. Area Freight and Fleet Management (FFM) review report, Brand (Brand Consult.), Rennesson (AFT-TNT)
9. Area Vehicle Control (VC) review report, Munck (Volvo) Glathe
10. Area Information and Communication Systems (IS/CS) review report, Pfliegl (Alcatel), von Pattay (ERTICO), Kossack (Siemens)
11. Area Contribution for Standardisation review report, von Pattay (ERTICO)

REFERENCIAS

[Abb90] *R. Abbott.* Resourceful Systems for Fault Tolerance, Reliability, and Safety. *ACM Computing Surveys. Vol 22, No. 1. 1990 pp. 35-68.*

[ADKM92] *Y. Amir, D. Dolev, S. Kramer, D. Malki.* Transis: A communication subsystem for high availability. *Proceedings of the 22nd International Symposium on Fault-Tolerant Computing. pp. 76-84. Julio 1992.*

[AG96] *B. Abernethy, J. Gunn.* A Different Perspective on National Architecture. *ITS Online, 1996*

[Ala96] *A. Alabau.* Comunicación personal. *Valencia, 1996*

[Aur96] *J.L. Aurtenetxe.* Aspectos legales y administrativos de la ordenación del tráfico. *I Congreso Internacional de Tráfico y Seguridad Vial en Euskadi. Servicio de Publicaciones del Gobierno Vasco, 1996.*

[Avi75] *A. Avizienis.* Fault Tolerance and Fault-Intolerance: Complementary Approaches to Reliable Computing. *Proc. Int. Conf. on Reliable Software. 1975 pp. 458-464.*

[Avi85] *A. Avizienis.* The N-Version Approach to Fault-Tolerant Software. *IEEE Transactions on Software Engineering. Diciembre 1985*

[BAH97] *Booz-Allen & Hamilton.* Current Status in Information and Communication Technology in Europe. *Bruselas, 1997*

[Ban94] *M. Banatre, P.A. Lee. ed.* Hardware and Software for Fault Tolerance. Experiences and perspectives. *Springer, 1994*

[Bar88] *R. Barnett, S. Maynard-Smith.* Packet Switched Networks. *Sigma Press, 1988*

[BBG89] *A. Borg, W. Blau, W. Graetsch, F. Herrmann, W. Oberle.* Fault Torerance under UNIX. *ACM Transactions on Computer Systems 7(1): 1-24. Febrero 1989.*

[BCS84] *D. Briatico, A. Ciuffoletti, L. Simoncini.* A distributed domino-effect free recovery algorithm. *Proceedings of the 4th Symposium on Reliable Distributed Systems. pp. 207-215. Octubre 1984.*

[Beau90] *K.G. Beauchamp.* Computer Communications. *2ª Edición. Chapman & Hall, 1990.*

[BJ87] *K.P. Birman and T.A. Joseph.* Reliable communication in the presence of failures. *ACM Transactions on Computer Systems, 5(1), Febrero 1987*

[Bla93] *U. Black.* Computer Networks. *2ª Edición. Prentice Hall. 1993.*

[Boy96] *A. Boyle.* Proyecto Piloto para la Autopista M-25 para el control de las autopistas. *I Congreso Internacional de Tráfico y Seguridad Vial en Euskadi. Servicio de Publicaciones del Gobierno Vasco, 1996.*

[Bri94] *O. Bridal.* Reliability Estimates for Repairable Fault-Tolerant Systems. *Nordic Seminar on Dependable Computing Systems. Lyngby, DK. 1994*

[Bri95] *O. Bridal.* A Methodology for Reliability Analysis of Fault-Tolerant Systems with Repairable Subsystems. *2nd IMA Conferenece on the Mathematics of Dependable Systems. York, UK. Septiembre 1995*

[Car96] *J. Carlson.* Comunicación personal por correo electrónico. *Xylogics Inc. 1996*

[CB94] *C.A. Cragg, B.L. Smith.* Intelligent Vehicle-Highway System (IVHS) Activities in the VDOT. *Virginia Transportation research Council, 1994.*

[Cer92] *V. Cerverón, C. Pérez.* Position Paper on the Interrupts Problem. *FASST internal document. 1992*

[Cer93a] *V. Cerverón, C. Pérez.* SIOP drivers. *FASST Technical Report. 1993*

[Cer93b] *V. Cerverón, C. Pérez, R. Martínez.* Tolerant: un driver para comunicaciones serie tolerante a fallos. *ACTA. Septiembre 1993*

[Cer96a] *V. Cerverón, G. Martín, V. Cavero.* Plataforma de desarrollo y experimentación de protocolos de comunicaciones. *II Jornadas de Informática. Granada. Julio 1996*

[Cer96b] *V. Cerverón, G. Martín, V. Cavero.* Modelo de Sistema de Comunicaciones Punto a Punto Tolerante a Fallos. *II Jornadas de Informática. Granada. Julio 1996*

[Chan75] *K.M. Chandy.* A Survey of Analytical Models of Rollback and recovery Strategies. *Computer, Mayo 1975*

[Che78] *L. Chen, A. Avizienis.* N-Version Programming: A Fault Tolerance Approach to Reliabilty of Software Operation. *Proceedings FTCS-8. Junio 1978.*

[Chen84] *C.L. Chen M.Y. Hsiao.* Error-Correcting Codes for Semiconductor Memory Applications: A State-of-the-Art Review. *IBM Journal of Research and Development, Marzo 1984*

[Cog97] *B. Coghlan.* Fault Tolerant Architectures Using Stable Storage Technology. *En preparación, 1997*

[Coh90] *S. Cohen.* Ingenierie du trafic routier. *Presses de l'ecole nationale des Ponts et chaussées, 1990*

[Come94] *D. Comer.* Internetworking with TCP/IP: Principles, Protocols, and Architecture. Tomos I, II y III. *Prentice Hall. 1991-1994.*

[Con95] *G.E. Conant.* Multilink PPP: One Big Virtual WAN Pipe. *Data Communications. Septiembre 1995*

[DM95] *E. Díaz, G. Martín.* Del factor humano a las nuevas tecnologías (capítulo 14). *Ed Síntesis, 1995*

[DMS96] *NTCIP Steering Group.* Dynamic Message Signs. *National Electrical Manufacturers Association, NEMA, 1996.*

[EC96] *European Commission. DG XIII.* Socio-economic impacts of telematics applications in Transport. Assessment of results from the 1992-1994 transport telematics projects. *Bruselas, 1996*

[Eln93] *E.N. Elnozahy.* Manetho: Fault Tolerance in Distributed Systems using Rollback-Recovery and Process Replication. *Tesis doctoral. Rice University (Houston, US), 1993*

[Equ96] *Equitel.* Sistema de Transmisión de datos entre estaciones remotas de control de tráfico. *1996.*

[ESS96] *NTCIP Steering Group.* Object Definitions for Environmental Sensor Station. *National Electrical Manufacturers Association, NEMA, 1996.*

[ET96] *EuroTrafiCom.* RTTI Information Exchange Standards for Euro-ISDN. *Proyecto EUROTRAFICOM - Deliverable 106 D1. 1996.*

[Fab96] *G. Fabregat.* Diseño de un Módulo de Proceso Tolerante a Fallos. Inclusión en una Arquitectura Multiprocesador. *Tesis Doctoral. Universitat de València. Febrero, 1996*

[Far95] *Farradyne Systems.* NTCIP Workshop White Paper. *Farradyne Systems, Inc, 1995*

[FASST90] *FASST Consortium.* FASST: Fault-tolerant Architecture with Stable Storage Technology. *FASST Project (ESPRIT P5212) Technical Annex, 1990*

[Fel97] *S. Felici.* Estudio, desarrollo y evaluación de protocolos para aplicaciones multimedia distribuidas sobre redes TCP/IP (Internet) para transferencia de información en tiempo real. *Tesis doctoral en preparación. Universitat de València, 1997*

[Fer96] *F. Fernández.* La gestión del tráfico en Madrid. *I Congreso Internacional de Tráfico y Seguridad Vial en Euskadi. Servicio de Publicaciones del Gobierno Vasco, 1996.*

[GTE96] *GTE Laboratories.* National ITS Architecture Communications Document. *National Arquitecture Research, 1996*

[Hech79] *H. Hecht.* Fault Tolerant Software. *IEEE Transactions on Reliability. Agosto 1979*

[Hop78] *A.L. Hopkins, Jr. T.B. Smith III, J.H. Lala.* FTMP A Highly Reliable Fault-Tolerant Multiprocessor for Aircraft Control. *Proceedings of the IEEE, Octubre 1978.*

[I-D BAP] *C. Richards, K. Smith.* PPP Bandwith Allocation Protocol. *(Internet Draft). IETF. Mayo 1996*

[I-D MP] *K. Sklower, B. Lloyd, G. McGregor, D. Carr, T. Coradetti.* The PPP Multilink Protocol. *IETF. Diciembre 1995.*

[Ins83] *A.D. Inselberg.* Multiprocessor Architecture ensures Fault-Tolerant Transaction Processing. *Mini-Micro Systems, Abril 1983.*

[ISO3309] *ISO/IEC.* Information technology -- Telecommunications and information exchange between systems -- High-level data link control (HDLC). procedures -- Frame structure. *ISO, 1993*

[ISO7478] *ISO.* Information processing systems -- Data communication -- Multilink procedures. *ISO, 1987*

[ISO7776] *ISO.* Information technology -- Telecommunications and information exchange between systems -- High-level data link control procedures-- Description of the X.25 LAPB-compatible DTE data link procedures *ISO, 1995*

[ITS96] *ITS Online.* Where is the National ITS System Architecture Effort Headed?. *ITS Online, 1996*

[Kal93] *Z. Kalbarczyk, J. Christmanson, H. Eldler.* Design Principles for Software Fault Tolerance - A survey. *Technical Report No.149 Chalmers University of Technology, SW. 1993*

[KHG] *M.K. Johnson.* The Linux Kernel Hackers' Guide, *1995*

[LAG] *O. Kirch.* The Linux Network Administrators' Guide. *1994*

[Lap85] *J. C.. Laprie.* Dependable Computing and Fault Tolerance: Concepts and Terminology. *Proc. of FTCS-15. 1985. pp. 1-11.*

[Lap92] *J. C. Laprie (ed.).* Dependability: Basic Concepts and Terminology. *Dependable Computing and Fault-Tolerant Systems series, Vol. 5. Spring-Verlag. 1992.*

[Lee90] *P. A. Lee, T. Anderson.* Fault Tolerance: Principle and Practice. *Second revised edition. Dependable Computing and Fault-Tolerant Systems. Vol. 3. Springer-Verlag. 1990.*

[Lor93] *Loral AeroSys.* Traffic Management Centers - The State of the Practice. *U.S. Departament of Transportation. ferderal Highway Administration, 1993.*

[Mar95] *J.J.Martínez.* Aplicaciones de vision artificial en la sensorización cualitativa del tráfico y la detección de incidentes en tiempo real. *Tesis doctoral. Universitat de València, 1995*

[Mar96] *G. Martín.* La situación europea de la tecnología. *El País, Sep'96*

[MIB] *K. McCloghrie* Management Information Base for Network Management of TCP/IP-based internets: MIB-II. *RFC 1213, 1991*

[Mic96] *E. Michelsen.* Comunicación personal por correo electrónico. *Copper Mountain Communications Inc, 1996*

[Mor93] PPP White Paper. *Morning Star Technologies. Agosto 1993.*

[Mul95] *G. Muller, M. Banâtre, M. Hue, N. Peyrouze, B. Rochat.* Lessons from FTM: an Experiment in the Design and Implementation of a Low Cost Fault Tolerant System. *Internal Report N° 913. IRISA, F. 1995*

[NAR96l] *National Arquitecture Research.* Logical Architecture. *Departamentro de Tráfico de los Estados Unidos de América, 1996*

[NAR96p] *National Arquitecture Research.* Physical Architecture. *Departamentro de Tráfico de los Estados Unidos de América, 1996*

[NAR96s] *National Arquitecture Research.* Executive Summary. *Departamentro de Tráfico de los Estados Unidos de América, 1996*

[Nel87] *V.P. Nelson, B.D. Carroll.* Tutorial: Fault-Tolerant Computing. *IEEE Computer Society Press, 1987*

[NPB96] *P. Nijkamp, G. Pepping, D. Banister.* Telematics and Transport Behaviour. *Springer, 1996.*

[NTCIP96a] *NTCIP Steering Group.* Class A Profile. *National Electrical Manufacturers Association, NEMA, 1996.*

[NTCIP96b] *NTCIP Steering Group.* Class B Profile. *National Electrical Manufacturers Association, NEMA, 1996.*

[NTCIP96c] *NTCIP Steering Group.* Class C Profile. *National Electrical Manufacturers Association, NEMA, 1996.*

[NTCIP96e] *NTCIP Steering Group.* Class E Profile. *National Electrical Manufacturers Association, NEMA, 1996.*

[NTCIP96f] *NTCIP Steering Group.* A family of protocols. *National Electrical Manufacturers Association, NEMA, 1996.*

[NTCIP96p] *NTCIP Steering Group.* Point to Multi-Point Protocol (PMPP). *National Electrical Manufacturers Association, NEMA, 1996.*

[NTCIP96s] *NTCIP Steering Group.* NTCIP Simple Transportation Management Framework. (T.S. 3.2-1996) *National Electrical Manufacturers Association, NEMA, 1996.*

[Obi96] *G. Obieta.* Sistemas de gestión de Tráfico. *I Congreso Internacional de Tráfico y Seguridad Vial en Euskadi. Servicio de Publicaciones del Gobierno Vasco, 1996.*

[Pas96] *F. Pastor.* Comunicación personal. *Electronic Traffic S.A, 1996*

[Pat95] *W.P. von Pattay.* Progress in Transport Telematics Standarisation. *CORD project- Deliverable D006. ERTICO, 1995.*

[Pat96] *W.P. von Pattay.* Standarisation for Intelligent Transport Systems (ITS). *Technical Report. ERTICO, 1996*

[Per97] *C. Pérez.* Micronúcleo para aplicaciones tolerantes a fallos. *Tesis doctoral en preparación. Universitat de València, 1997*

[Pov96] *R. Povel.* Generaciones Futuras de Vehículos Comerciales. *I Congreso Internacional de Tráfico y Seguridad Vial en Euskadi. Servicio de Publicaciones del Gobierno Vasco, 1996.*

[Pow94] *B. Powell.* Lost on the infobahn?. *Newsweek, Oct 1994*

[Ran75] *B. Randell.* System Structures for Software Fault Tolerance. *IEEE Transactions on Software Engineering. Junio 1975*

[Rand96] *D. Rand.* Comunicación personal por correo electrónico. *Bungi, 1996*

[Ren78] *D.A. Rennels, A. Avizienis.* A Study of Standard Building Blocks for the Design of Fault-Tolerant Distributed Computer Systems. *Proc.FTCS-8 Junio 1978*

[RFC 1122] *R. T. Braden* Requirements for Internet hosts - communications layers. *Octubre 1989.*

[RFC 1332] *G. McGregor.* The PPP Internet Protocol Control Protocol (IPCP). *IETF. Mayo 1992.*

[RFC 1333] *W. Simpson.* PPP Link Quality Montoring. *IETF. Mayo 1992.*

[RFC 1334] *B. Lloyd, W. Simpson.* PPP Authentication Protocols. *IETF. Octubre 1992.*

[RFC 1547] *D. Perkins.* Requeriments for a Internet Standard Point-to-Point Protocol. *Diciembre 1993.*

[RFC 1570] *W. Simpson.* PPP LCP Extensions. *IETF. Enero 1994*

[RFC 1661] *W. Simpson.* The Point-to-Point Protocol (PPP). *IETF. Julio 1994*

[RFC 1662] *W. Simpson.* PPP in HDLC-like Framing. *IETF. Julio 1994*

[RFC 1663] *D. Rand.* PPP Reliable Transmission. *IETF. Julio 1994*

[RFC 1717] *K. Sklower, B. Lloyd, G. McGregor, D. Carr.* PPP Multilink Procedure. *IETF. 1994.*

[Rif92] *J.M. Rifflet.* Comunicaciones en UNIX. *McGraw-Hill, 1992*

[Rifa91] *J. Rifà, Ll. Huguet.* Comunicación Digital. Teoría Matemática de la Información. Codificación Algebraica. Criptología. *Ed. Masson, 1991*

[Rifa95] *J. Rifà.* Seguridad Computacional. *Servicio de publicaciones de la Universitat Autònoma de Barcelona, 1995.*

[RKR96] *I. Routledge, S. Kemp, B. Radia* UTMC: The way forward for Urban Traffic Control. *Traffic Engineering + Control, Vol 37 No 11. Noviembre 1996*

[RMD97] *R. Martínez.* Evaluación de sistemas tolerantes a fallos: análisis de cobertura de fallos. *Tesis doctoral en preparación. Universitat de València, 1997*

[RP20] *M. Callahan, A. Longyear.* PPP for Linux. *On-line documentation. 1994*

[Sch96] *V. Schryver.* Comunicación personal por correo electrónico. *Denver, 1996*

[SET93] *SETRA.* Réseaux de télétransmission des autoroutes de liaison non concedées. Dispositions techniques générales. *Service d'études techniques des routes et autoroutes, 1994.*

[SET94] *SETRA.* Panneaux de signalisation à messages variables. *Service d'études techniques des routes et autoroutes, 1994.*

[Sny95] *J. Snyder.* Rethinking the NTCIP Design and Protocols - Analyzing the Issues. *Opus One, 1995*

[Sta94] *W. Stallings.* Data and Computer Communications. *4ª Edición. Macmillan Publishing Company. 1994*

[Sutt91] *R. Sutterfield.* Low-Cost IP Connectivity. *Diciembre 1991.*

[Swe91] *P. Sweeney.* Error Control Coding. An Introduction. *Prentice Hall. 1991.*

[Tan91] *A. Tanenbaum.* Redes de Ordenadores. *2ª Edición. Prentice Hall. 1991.*

[Tan96] *A. Tanenbaum.* Computer Networks. *Prentice Hall. 1996.*

[Tay80] *D.J. Taylor, D.E. Morgan, J.P. Black.* Redundancy in Data Structures: Improving Software Fault Tolerance. *IEEE Transactions on Software Engineering. Noviembre 1980*

[TTI94] *Texas Transport Institute* Informe sobre impacto socioeconómico de la movilidad. *1994*

[WaDoT96] *Departamento de Tráfico del Estado de Washington.* Aplication of Advanced Transportation Technology Within Washington State. Discussion and Policy Recommendations. *ATTP Commission, 1996.*

[Wer96] *J. Werner.* Inside San Antonio TransGuide. *ITS Online, 1996.*

[Ying80] *Ying W. NG, A.A. Avizienis.* A Unified Reliability Model for Fault-Tolerant Computers. *IEEE Transactions on Computers, Noviembre 1980*

[Wic95] *S. Wicker.* Error Control System for Digital Communication and Storage. *Prentice Hall, 1995*

LISTADO DE ACRÓNIMOS

ACCM	Asynchronous Control Character Map
ARQ	Automatic ReQuest for replay
ATIS	Advanced Transport Information Systems
ATMS	Advanced Traffic Management Systems
ATT	Advanced Transport Telematics
AVCS	Advanced Vehicle Control Systems
AVI	Automatic Vehicle Identification
CEN	Comité Europeo para la Normalización
CRC	Cyclic Redundant Code
CTU	Control de Tráfico Urbano
DAI	Detección Automática de Incidentes
DSRC	Dedicated Short Range Communications
EDI	Electronic Data Interchange
EOF	Equipo de Observancia Forzosa
ERTICO	European Road Transport telematics Implementation Coordination Org.
ETD	Estación de Toma de Datos
EVA	Estación de Visión Artificial
FASST	Fault-tolerant Architecture for Stable Storage Technology
FCS	Frame Check Sequence (=SVT)
FEC	Forward Error Control
FHWA	Federal HighWay Administration

GSM	Global System for Mobile communications
HDLC	High Data Link Control
IP	Internet Protocol
IPCP	Internet Protocol Control Protocol
ISO	International Standards Organization
ITS	Intelligent Transport Systems
IVC	Indicador de Vía Controlada
LAN	Local Area Network
LAPB	Link Access Control Balanced
LCP	Link Control Protocol
LQP	Link Quality Protocol
LQR	Link Quality Report
MAN	Metropolitan Area Network
MLP	dispositivo que implementa el MLPPP en esta memoria
MLPPP	MultiLink Point to Point Protocol
MRU	Maximum Receive Unit
MTU	Maximum Transmission Unit
NCP	Network Control Protocol
NMR	N-Modular Redundancy
NTCIP	National Traffic Control IVHS communication Protocol
OSI	Open System Interconnection
PDN	Public Data Networks
PDU	Protocol Data Unit
PMPP	Point to MultiPoint Protocol
PMV	Panel de Mensajes Variables
PPP	Point to Point Protocol
RDS-TMC	Radio Data System - Traffic Message Channel
RDSI	Red Digital de Servicios Integrados (=ISDN)

RFC	Request For Comments
RTI	Road Transport Informatics
SLIP	Serial Line Internet Protocol
STMF	Simple Traffic Management Framework
SVT	Secuencia de Verificación de Trama (=FCS)
TC	Technical Committee
TCP	Transport Control Protocol
TMR	Triple Modular Redundancy
UDP	User Datagram Protocol
USDOT	United States Department of Transport
UTMC	Urban Traffic Management and Control
VAO	Vehículo de Alta Ocupación (=HOV)
WAN	Wide Area Network
WG	Working Group

UNIVERSITAT DE VALÈNCIA FACULTAD DE CIÈNCIES FÍSIQUES

Reunit el Tribunal que subscriu, en el dia de la data, acordà d'atorgar, ~~per unanimitat~~, a aquesta Tesi Doctoral d'En/ Na/ N' VICENTE CERVERON LLEÓ

la qualificació de APTO CUM LAUDE

València a 10 de FEBRERO de 1997

El Secretari,

El President,

UNIVERSITAT DE VALÈNCIA
BURJASSOT
FACULTAT DE CIÈNCIES FÍSIQUES