

## Bidireccionalidad y eficiencia en el transporte de datos de teleoperación a través de redes IP

M. Diaz-Cacho, A. Barreiro y M. G. Rivera

Departamento de Ingeniería de Sistemas y Automática, Universidad de Vigo,  
E.T.S. Ingenieros Industriales. 36200-Vigo, España  
(e-mail: mcacho@uvigo.es, abarreiro@uvigo.es, mgrivera@uvigo.es)

**Resumen:** Este artículo propone un nuevo esquema de protocolo de transporte que aprovecha las particularidades de flujos de datos de teleoperación a través de Internet. Se presenta como resultado de un análisis global de los datos de la teleoperación bilateral para ser encapsulados en paquetes compatibles IP. Distingue entre tráfico multimedia, tráfico supermedia y tráfico de control, y ofrece la posibilidad de sistemas de control del flujo que sean amigables con el tráfico TCP (*TCPFriendly*) mayoritario en Internet. Se presentan resultados de simulación y comparaciones con otros esquemas, utilizando el sistema de control del flujo trinomial, pero con posibilidad de otros sistemas de control. El esquema propuesto aprovecha la bidireccionalidad del lazo de control y el pequeño tamaño de los datos de teleoperación para la mejora de la eficiencia de la transmisión manteniendo la misma información a enviar, y pretende servir de aportación por parte de la red, a los esfuerzos por modelar la estabilidad de sistemas de teleoperación a través de redes con retardos variables mediante las actuaciones sobre los controladores maestro y esclavo. En el artículo se presenta una cabecera de transporte adecuada a los sistemas de control del flujo basados en ecuación (*Equation-Based Flow Control*) y se establecen unos usos de determinados campos de la cabecera actual de Internet. *Copyright © 2010 CEA.*

**Palabras Clave:** Teleoperation, Protocols, Transport delay, Simulation, Flow control, Network controlled systems, Data flow analysis, Headers, Telerobotics.

### 1. INTRODUCCIÓN

Un gran número de estudios de teleoperación y de sistemas de control en red NCS (*Network Controled Systems*) se concentran fundamentalmente en la adaptación de los parámetros de estabilidad y pasividad de las variables de onda a redes con retardos y con pérdida de paquetes (Anderson and Spong, 1988), (Niemeyer and Slotline, 1991). Las mejoras encontradas en los casos en los que se implementen retardos constantes en las redes animaron a su ampliación a las redes con retardos variables (Munir and Book, 2001). El fundamento de estas líneas de investigación está basado en la posibilidad del uso de Internet como medio de comunicación entre los dispositivos que componen un entorno de teleoperación. Una plataforma de teleoperación bilateral debe disponer de mínimamente dos flujos de datos, básicamente independientes entre sí y con envío de información en sentidos contrarios. El dispositivo de mando, operado o movido directamente por un ser humano suele ser un *joystick* u otra interface hombre-máquina conocidas como *HSI* (*Human System Interface*) y se le llama dispositivo maestro o *Master*, mientras que el dispositivo teleoperado, habitualmente de tipo mecánico suele llamarse dispositivo esclavo o *Slave*. El flujo de datos desde el maestro hacia el esclavo lleva información de operación sobre los actuadores del esclavo, y el flujo inverso desde el esclavo hacia el maestro lleva información de estado, que suele ser del tipo háptico, visual u otros y que sirve al operador para cerrar el bucle de control. En algunos trabajos (Cen *et al.*, 2005) se llama a los datos de teleoperación “datos supermedia”, para distinguirlos de los “datos multimedia” con

los que tienen algunas características diferenciales. En este trabajo se llamará “datos supermedia” a los datos de tipo posición y retornos hápticos o de estado, en general datos de un tamaño reducido y que deberían ser entregados a la aplicación lo mas rápidamente posible en lugar de encolarse en un *buffer*. Por contra se seguirá llamando “datos multimedia” a los referidos a envíos o retornos de vídeo y audio. Finalmente se distinguen los “datos de control”, referidos a cambios de configuración u operaciones en las que se prima la seguridad de la entrega sobre la rapidez de la misma. En la figura 1 se representa la teleoperación coordinada tipo SMMS (*Single-Master Multi-Slave*) de una grúa tipo pórtico con seguimiento automático de cámara y retornos háptico de grúa, de vídeo y de estado (Díaz-Cacho *et al.*, 2008), (Fernandez Villaverde *et al.*, 2007).

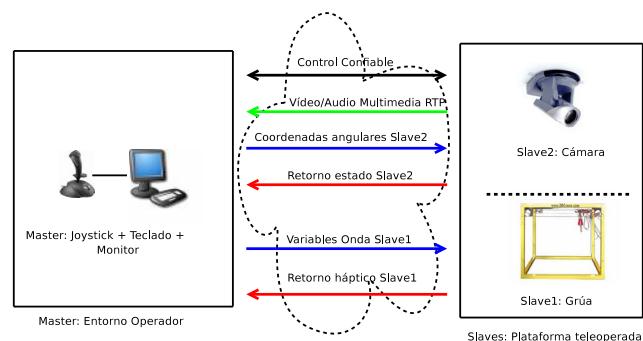


Figura 1. Canales de teleoperación

Las líneas de investigación en NCS abordan dos campos principales, por un lado los trabajos sobre los sistemas de control o controladores maestro y esclavo, para disminuir los efectos de los retardos en la estabilidad de los sistemas, y por otro lado, que es el enfoque que sigue este artículo, los esfuerzos por modelar el tipo de tráfico que la teleoperación produce y la búsqueda de protocolos, dispositivos o estructuras de red que lo optimicen y permitan acercarse a los requisitos que estos sistemas tienen, basados en las características de tiempo real y estabilidad. Existe una línea intermedia, a veces llamada de soluciones *Middleware*, en las que se propone adaptar los datos y la información a los requisitos del sistema de teleoperación antes de depositarlos en la red. Soluciones *Middleware* son los esquemas de mantenimiento de pasividad (Berestesky *et al.*, 2004), organización de Ganancia o *Gain Scheduling* (Tipsuwan and Chow, 2004), control con “Banda Muerta” (*Deadband Control*) (Hirche *et al.*, 2005), compensación del retardo (Emanuel Slawinski, 2006), (Emanuel Slawinski and Mut, 2007) u otros.

En general, cualquier sistema de teleoperación tiene unas particularidades que lo distinguen de otros sistemas en red. A resaltar:

- La teleoperación es bilateral, es decir, existe un canal de retorno de datos, y no sólo de asentimientos o estados de la red, habitual en protocolos de transporte. A los paquetes de retorno de tipo asentimiento se les llama paquetes ACK.
- Requisitos de tiempo real. Estos requisitos son necesarios para mantener la estabilidad del sistema. Los estudios al respecto lo acotan a requisitos de retardos máximos y tasa máxima de pérdida de paquetes (García-Rivera and Barreiro, 2007).
- Tamaño de los datos generalmente muy pequeño, del mismo orden de magnitud que los paquetes tipo ACK de los protocolos de transporte. Al mismo tiempo, este tamaño suele ser fijo para los “datos supermedia” de teleoperación en tiempo real. Se han encontrado propuestas que conllevan campos de datos entre 20 y 100 octetos (Ping *et al.*, 2005), (Rejaie *et al.*, 1999), y en contados casos de tamaños mayores, aunque no se suele especificar ni justificar (Uchimura and Yakoh, 2004). Cuando el retorno es de tipo vídeo, el tamaño de los datos aumenta, y las características del retorno pueden cambiar, pasándose a datos de tipo “datos multimedia” y existiendo algunos protocolos que resuelven sus requisitos, fundamentalmente los derivados de RTP (Group *et al.*, 1996), (Schulzrinne *et al.*, 2003), pero cuyas características de tiempo real son menos estrictas que con los “datos supermedia”, ya que en (Kopetz, 1997) se sugieren tamaños de datos pequeños para este tipo de sistemas.

Si la teleoperación se realiza a través de una red pública como Internet, el tráfico mayoritario es el tráfico TCP (Transmission Control Protocol), y el diseño de nuevos protocolos de transporte tiene que cumplir requisitos de *TCPFriendly*, *Fairness* y *Best Effort* (Floyd and Fall, 1999). Un flujo de Internet se dice que es *TCPFriendly* si consume aproximadamente el mismo ancho de banda que un flujo TCP sobre la misma ruta y en las mismas condiciones de retardo y pérdida de paquetes, en un espacio de tiempo determinado. Esto intenta mantener los protocolos de transporte al mismo nivel que el protocolo mayoritario para no canibalizarlo, al disponer este de un sistema de control del flujo que se adapta a la congestión de la red (Floyd and Fall, 1999).

*Fairness* implica que todos los flujos de datos que utilicen ese protocolo tendrán las mismas posibilidades de disponer de ancho de banda para su uso. La característica *Best Effort* implica que se utilizará, si es necesario, todo el ancho de banda ofrecido por la red, y ese uso debe ser dinámico, es decir, que se adapta a los cambios de disponibilidad de ancho de banda extremo a extremo provocados por su utilización.

Los protocolos de transporte más utilizados, TCP y UDP (User Datagram Protocol) no pueden cumplir con esas premisas, pues TCP es un protocolo orientado a conexión, confiable, pero con un *jitter* grande y con penalización del retardo a favor de la retransmisión de paquetes detectados como perdidos. TCP además dispone de una cabecera de 20 octetos, que representa una baja eficiencia de la carga útil al ser esta de un tamaño similar al de los datos que contiene. UDP es un protocolo no orientado a conexión y no confiable, que puede cumplir los requisitos de tiempo real e incluso requisitos de eficiencia del tamaño de la cabecera respecto al tamaño de los datos, pero que no cumple los requisitos de *Fairness* y *TCPFriendly*. Alguno de los protocolos multimedia que van sobre el transporte UDP tienen grandes problemas para cumplir características *TCP-Friendly* y *Fairness*, y su uso masivo puede provocar un colapso en Internet (Floyd and Fall, 1999), (Rhee *et al.*, 2000).

La teleoperación bilateral permite aprovechar el canal de retorno en cualquiera de los sentidos para el envío de paquetes de asentimiento ACK con información de estado de la red, basada en cálculos de tiempos y números de secuencia. Básicamente, los paquetes ACK sirven para la algoritmia de control de flujo basado en origen (*source based flow control*), o para ofrecer servicios de confiabilidad y recuperación ante pérdidas. En los esquemas de protocolos de teleoperación clásica, los paquetes ACK conforman un flujo de paquetes independiente del flujo de datos de teleoperación. El aprovechamiento propuesto del flujo de datos de retorno para el control del flujo en origen puede visualizarse en la figura 2, donde la información ACK forma parte de la cabecera de los paquetes de datos.

Este artículo está organizado en 6 secciones. La primera sección es la sección introductoria donde se presenta la motivación y una primera visión de la propuesta. La segunda sección es un estado del arte. La tercera sección es una presentación del esquema propuesto, donde se muestra la bidireccionalidad de los datos y cómo aprovecharla y a su vez un esquema de control del flujo basado en el esquema trinomial. La cuarta sección es una propuesta de cabecera de transporte, y un análisis de la eficiencia de los paquetes de teleoperación. La quinta sección presenta resultados comparativos de simulación sobre ns2 del esquema propuesto y finalmente en la sexta sección se ven las conclusiones y se plantean líneas futuras de ampliación, trabajo y aplicación de resultados.

## 2. ESTADO DEL ARTE

Existen en la actualidad abiertas múltiples líneas de investigación en protocolos de teleoperación. Los autores han identificado tres líneas básicas de trabajo, clasificadas en función de la capa del modelo OSI/TCP/IP donde se aplican: protocolos de enlace, protocolos de transporte y protocolos de aplicación. En la figura 3 se presenta un resumen esquemático del estado del arte en protocolos de teleoperación. Algunas de las soluciones presentadas en la figura están todavía en fase de definición, son muy genéricas o no disponen de mayores artículos que muestren su funcionamiento concreto y sus prestaciones.

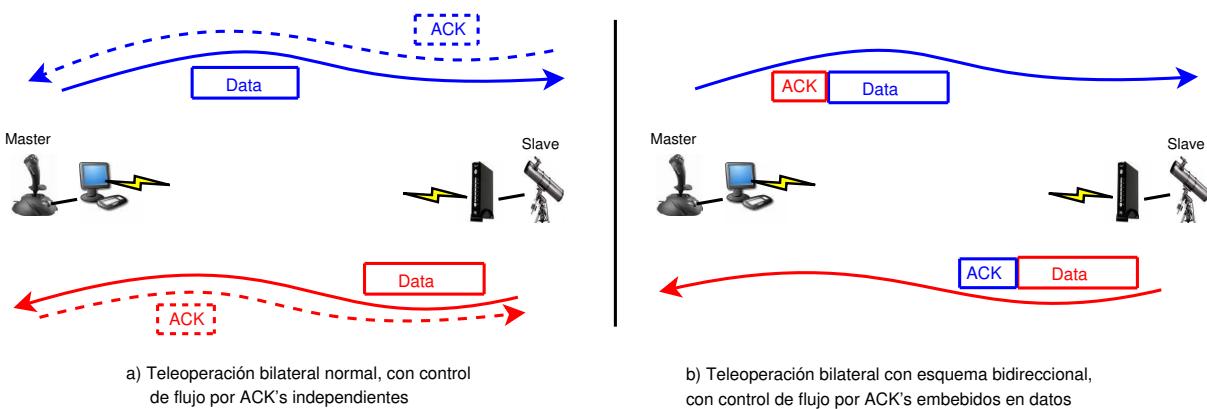


Figura 2. Canal de retorno de datos y estado de la red



Enlace	CAN, Profibus, RTE, RTNP ZigBee, Bluetooth, 802.11	ATM, MPLS, ISDN, FR	CAN, Profibus, RTE, RTNP ZigBee, Bluetooth, 802.11
Red	IPv4, IPv6, VAN, QoS, Multicast, RED-Gateways		
Transporte	Trinomial, RAP, TEAR, IRTP, TFRC, TFRC-Probe, DCCP, ETP, SNRTP, RTP		
Aplicación	SNRP, FrameWorks, STRON, RON, QRON, OverQoS		

Figura 3. Estado del arte en protocolos de teleoperación

## 2.1 Capa de enlace

La capa de enlace cubre los requisitos de transmisión entre dos nodos que están directamente conectados entre sí. La propia naturaleza de Internet permite presentar la topología de un sistema de teleoperación con redes de acceso del maestro y del esclavo, y la red troncal de Internet que une ambas, como se puede ver en la figura 3. Las tecnologías de enlace de las redes de acceso son accesibles y configurables por los usuarios, y pueden ser escogidas de entre múltiples disponibles o incluso desarrollar la propia (Uchimura and Yakoh, 2004). Las tecnologías de enlace de Internet requieren de grandes consensos para su modificación o sustitución. Esto, unido al hecho de las altas prestaciones que dan las tecnologías troncales ATM, MPLS, y en menor medida MetroEthernet, permitiendo requisitos de QoS suficientes para la transmisión supermedia y multimedia, hace que las líneas abiertas para la creación de protocolos de enlace específicos para teleoperación a ser utilizados por la troncal de Internet sean prácticamente inexistentes.

RTNP (Real Time Network Protocol) (Uchimura and Yakoh, 2004) y (Yutaka Uchimura and Ohnishi, 2005) propone una nueva arquitectura de protocolos a ser interpretada por el sistema operativo, que debería ser de tiempo real para que fuese totalmente efectiva. El corazón del protocolo consiste en la inserción de una etiqueta en la cabecera Ethernet, que identifique una trama de alta prioridad para los requisitos de tiempo real de la aplicación de teleoperación. Esa etiqueta es interpretada

por la nueva capa de enlace que lleva la trama directamente a la capa de aplicación. En la capa de aplicación se interpreta la interrupción y se actua en consecuencia, sin pasar por las colas habituales de la capa Ethernet ni esperar turno en los procesos de la capa de red. El protocolo necesitaría que se modificasen los sistemas comutadores intermedios, pues estos, al analizar los campos de la cabecera Ethernet, no reconocerían la cabecera RTNP. En cambio tendría eficacia en transmisiones punto a punto de la red local distribuida, con concentradores hub como elementos distribuidores, al no analizar estos la cabecera de la capa de enlace. También podría hacerse compatible con las tecnologías VLAN 802.1q.

En las soluciones de capa de enlace, se centran esfuerzos por parte de la industria y de la comunidad científica en la tecnología Ethernet. Soluciones de la capa de enlace ya implementadas y algunas con dispositivos en el mercado son las derivadas de aplicar características de tiempo real a las redes Ethernet. Estas forman el conjunto RTE (Real Time Ethernet) que básicamente tratan de aprovechar la infraestructura Ethernet para soluciones de buses de campo. La propuesta de unificación de estas tecnologías por parte de la industria está centrada en la creación de un estándar (Felser, 2005), (Decotignie, 2005), (Schiffer, 2001). De forma resumida, las soluciones tratan de identificar tramas con requisitos de tiempo real mediante la inserción de identificadores en la cabecera Ethernet, y que los comutadores intermedios prioricen las tramas según los indicadores de prioridad insertados, o que directamente dis-

pongan de canales físicos reservados para el envío de estos paquetes mediante técnicas TDMA (*Time Division Multiple Acces*). Los sistemas de prioridades se suelen considerar semi-deterministas, al poder predecir características de tiempo real en un rango de retardo del enlace, y los sistemas de reserva de canal se consideran deterministas, al no existir medio compartido, ni por tanto competencia por el medio ni sistema de prioridades.

Otras soluciones de capa de enlace son las utilizadas para medios inalámbricos. En los medios inalámbricos, la probabilidad de pérdida o error de paquete es mucho mayor, lo que penaliza el *jitter* de los protocolos confiables, o penaliza la tasa de pérdidas de los no confiables. Son de destacar soluciones comerciales como ZigBee 802.15.4 orientada a las redes de sensores-actuadores con características de autoconfiguración, Bluetooth orientada mas a la interconexión de periféricos hacia un sistema central, o las soluciones Wireless Lan 802.11(a/b/g) donde la acción de compensación de pérdidas o *jitter* se realiza en la capa de transporte.

## 2.2 Capa de transporte

Es en la capa de transporte donde mas interés hay para la mejora de los servicios de teleoperación, y también donde mayores trabajos se han realizado, fundamentalmente en dotar a los protocolos de sistemas de control del flujo. Las soluciones encontradas se presentan en la figura 3, y se comentan los protocolos RAP, TEAR, Trinomial, TFRC, ETP e IRTP. No obstante, posiblemente el protocolo de transporte mas utilizado en teleoperación a través de Internet sea UDP, pero UDP no cumple con las normas de tráfico en Internet, como lo son las características *TCPFriendly* y *Fairness*. Existen diversos intentos de dotar a UDP de sistemas de control de la congestión, entre los que cabe destacar el DCCP (*Datagram Congestion Control Protocol*) (Kohler *et al.*, 2006b), (Kohler *et al.*, 2006a).

El protocolo RAP (*Rate Adaptation Protocol*) (Rejaie *et al.*, 1999) dispone de un algoritmo de control del flujo AIMD (*Additive Increase-Multiplicative Decrease*) de tipo dinámico, que se actualiza cada retardo de ida y vuelta o RTT (*Round Trip Time*). Este RTT es estimado mediante los ACK devueltos por cada paquete, y permite observar el estado de la red sobre los paquetes enviados en el paso anterior, y por tanto actualizar la velocidad del flujo en el siguiente paso. Presenta el concepto de cluster de pérdidas, que modela la pérdida de paquetes mediante ráfagas, ignorando las pérdidas si se producen dentro del Cluster, al haberse ya activado el sistema de control del flujo en la detección de la primera pérdida. Esto permite tener funcionalidades parecidas al protocolo TCP, pero con mejores prestaciones en lo que se refiere al sistema de recuperación de pérdidas, mejorando el rendimiento. Este protocolo fué el punto de partida del protocolo Trinomial (Liu *et al.*, 2002).

El protocolo TEAR (*TCP Emulation At Receivers*) (Rhee *et al.*, 2000) parte de la suposición de que la probabilidad de pérdida de paquetes es independiente de la velocidad de transmisión. Por lo tanto, no se necesita recalcular la velocidad de transmisión en el emisor cada vez que este detecta una pérdida, y en su lugar puede calcularse en el receptor cada vez que este último conoce el estado de la red y la velocidad mas adecuada a la que puede recibir los datos. Básicamente aumenta la ventana de tiempo para retornar el estado de la red al emisor, para disminuir la variación del rendimiento. En TCP, el retorno es por paquete, y en TEAR se retorna con un periodo mas amplio,

lo que mantiene constante la tasa de transmisión mas tiempo, disminuyendo el *jitter* provocado por el control del flujo. El control del flujo pasa a depender del receptor, y por tanto el modelado se hace en unidades de retardo ascendente, en lugar de RTT. El tipo de paquetes para el que da mejores resultados este protocolo es para paquetes de tamaño medio/alto, sin recuperación ante pérdidas y con requisitos de *jitter* de latencia mas estrictos que TCP normal o derivados TFRC. La adaptación de cambios de velocidad de transmisión para paquetes de tamaño pequeño es muy lenta, y no tiene prácticamente repercusión en el rendimiento del protocolo. Sus características principales hacen que sea adecuado para ambientes asimétricos, donde la disponibilidad de ancho de banda es diferente en ambos sentidos de transmisión, y para flujos de paquetes de tamaño superiores a los 1000 octetos.

El protocolo trinomial (Liu *et al.*, 2002), está mas definido como algoritmo de control de flujo por sus autores que como protocolo en sí, al describir básicamente el procedimiento de control del flujo y el proceso de envío de datos y asentimientos ACK. No entra al detalle del formato de la cabecera, ni de los requisitos de direccionamiento de transporte, por lo que puede implementarse sobre una cabecera UDP para darle mas funcionalidades, o puede crearse una cabecera específica. En este trabajo se ha adoptado este algoritmo de control del flujo sobre una definición de cabecera propia, orientada a mejorar la eficiencia de estos protocolos. La descripción del algoritmo se realiza en la sección 3.2.

El sistema TFRC (*TCP-Friendly Rate Control*) (Floyd *et al.*, 2000), (Handley *et al.*, 2003), es un mecanismo basado en TCP pero con menor variación de rendimiento (*throughput*) que este último, lo que lo hace mas adecuado para aplicaciones de telefonía, *streaming* o multimedia en general. Realmente es un mecanismo de control de congestión, para entornos de Internet y con características *TCPFriendly*. Al igual que el TEAR, es un control basado en el receptor, que utiliza el tiempo entre pérdidas para modelar la velocidad de transmisión informando al emisor mediante ACK's. Este protocolo es habitual en comparaciones para protocolos de teleoperación, con variantes para tráfico del tipo VoIP (Floyd and Kempf, 2004) o para paquetes pequeños TFRC-SP (Floyd and Kohler, 2007). Este mecanismo ha derivado en la creación del protocolo DCCP para dotar a los datagramas tipo UDP de control del flujo (Kohler *et al.*, 2006a), y ha servido de punto de partida para la toma de conciencia de las necesidades futuras de una Internet multimedia con la creación de nuevas técnicas. Las variantes de esta suite de mecanismos y protocolos deberán ser estudiadas y posiblemente adaptadas a los mecanismos propuestos en este trabajo, como se plantea en la sección 6.

Los protocolos ETP (*Efficient Transport Protocol*) (Wirz *et al.*, 2008a), SNRTP (*Simple Network Robot Transport Protocol*) y SNRP (*Simple Network Robot Protocol*) (Wirz *et al.*, 2007) (Wirz *et al.*, 2008b), por los mismos autores, intentan establecer una arquitectura de protocolos que abarquen las capas de transporte y aplicación, con el desarrollo de entornos de trabajo para sistemas MMMS (*Multi-Master Multi-Slave*). ETP es útil en el caso en que los requisitos de ancho de banda son menores que los requisitos de retardo y pérdidas. En ese caso puede ser interesante su estudio como complemento al presentado en este trabajo, ya que también es bastante específico para teleoperación háptica o de datos de pequeño tamaño en ráfagas de información relativamente constante. Básicamente establece seis estados para el control del flujo, con lo que consigue estabilizar

el flujo evitando grandes fluctuaciones del rendimiento. Estos protocolos y entornos de trabajo están en fase de definición.

El protocolo IRTP (*Interactive Real-Time Protocol*) (Ping *et al.*, 2005) analiza la eficiencia de protocolos de transporte para teleoperación bilateral donde el tamaño de los datos vuelve a ser pequeño, centrándose en la reducción del tamaño de la cabecera y mejorar la eficiencia. Se compara con el protocolo UDP, manteniendo un tamaño de cabecera similar, pero mejorando sus funcionalidades, dotándolo de recuperación ante pérdidas y control de flujo. Este protocolo permite distinguir dinámicamente datos de control de datos de teleoperación o supermedia. Es un protocolo orientado a conexión.

### 2.3 Capa de Aplicación

Las soluciones en capa de aplicación son muy diversas y no se contemplan profusamente en este trabajo. Habitualmente están centradas en el desarrollo de *frameworks* o entornos de trabajo para topologías MMMS específicas, definiendo entornos de acceso a los datos Web o Java como SNRP (Wirz *et al.*, 2008a), o formatos de campos de datos. Existe una línea de soluciones interesante sobre redes superpuestas u *Overlay* que se presenta a continuación destacándose el protocolo STRON.

El planteamiento de STRON (*Supermedia Transport for teleoperation over Overlay Networks*) (Cen *et al.*, 2005) es el de construir una red *Overlay* específica para teleoperación, de forma que se implemente un algoritmo de enrutamiento que minimice las pérdidas y el retardo a través de enlaces lógicos. Existen diversas soluciones sobre redes *Overlay*, a destacar OverQoS (Subramanian *et al.*, 2004), OverCast for Multicast (Jannotti *et al.*, 2000), RON (Andersen *et al.*, 2001), QRON (Li and Mohapatra, 2004) y SON (Duan *et al.*, 2003).

Las redes *Overlay* son una buena alternativa pues las propuestas previas para el establecimiento y control de una calidad de servicio QoS como lo son IntServ, DiffServ, e IP Multicast no han tenido una aceptación muy amplia ya que requieren de la modificación de todos los enruteadores de la red. Las soluciones *Overlay* son en principio compatibles con otras soluciones de capa de transporte.

## 3. ESQUEMA PROPUESTO

En este trabajo, se presenta una propuesta de transmisión de “datos supermedia”, llamada TDTP (*Teleoperation Data Transport Protocol*), donde los requisitos de pérdida de paquetes y retardos son los que determinarían los límites de estabilidad del sistema de teleoperación asociado (García-Rivera and Barreiro, 2007). El protocolo se centra por tanto en cumplir con nuevos requisitos de uso de la red, como el *TCPFriendly*, que prevé un reparto equitativo de los recursos de la red por parte de todos los flujos que la usen. Si hubiese varios flujos de datos tipo supermedia en los enlaces de Internet, los protocolos sin control del flujo como UDP, llenarían las colas de los enlaces de Internet y tendrían un efecto negativo y descontrolado sobre su rendimiento, sus pérdidas y el retardo de sus paquetes. Así, es necesario prever esquemas y protocolos que dispongan de mecanismos de control del flujo y que al mismo tiempo optimicen los recursos. En este trabajo se ofrecen resultados con el mecanismo trinomial y la optimización de recursos en la disminución de la cantidad de paquetes a enviar manteniendo las mismas prestaciones mediante la bidireccionalidad. La implementación del control del flujo implica una disminución del

rendimiento respecto a esquemas sin control del flujo, pero en cambio se produciría una disminución de las pérdidas, y mayores probabilidades de disminución del retardo. Si el control del flujo sigue pautas *TCPFriendly*, se asegura que en largas ventanas temporales, el tráfico supermedia compite al menos en igualdad de condiciones que el tráfico TCP por el uso de los canales compartidos.

TDTP está diseñado específicamente para teleoperación con retorno de tipo haptico, pues son datos de pequeño tamaño con información de fuerza, velocidad, posición o captaciones de sensores específicos como temperatura o distancia a obstáculos. En la práctica, el protocolo es compatible con otros tipos de datos como audio o vídeo en tiempo real, ya sea de forma directa o mediante el uso de otros canales de transporte. El esquema TDTP se basa en el aprovechamiento de la bidireccionalidad, en el control del flujo y en la cabecera del protocolo. En la figura 4 puede verse un diagrama de operación.

El control del flujo propuesto se basa en el conocimiento de la pérdida de paquetes mediante el análisis de los números de secuencia en recepción y el envío de un dato booleano informando de si ha habido pérdida o no en el flujo de retorno. Para ello se dispondrá al menos de campos de número de secuencia y un campo de un bit que informa de la pérdida de datos, llamado BCN (*Backward Congestion Notification*).

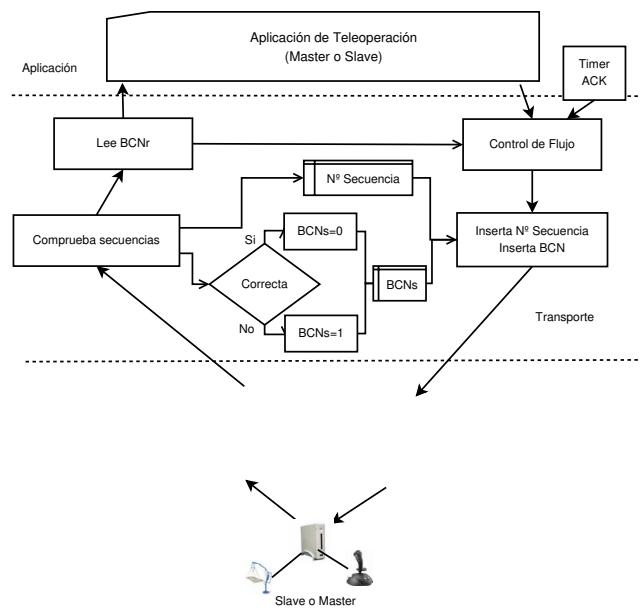


Figura 4. Esquema general

El diagrama tiene dos hilos de funcionamiento, el hilo de transmisión de datos y el hilo de recepción. Ambos hilos funcionan en los dispositivos maestro y esclavo. El primero recibe los “datos supermedia” de la capa de aplicación, ya sea directamente de los dispositivos o sensores, o de la interfaz humana maestra, y los envía a la capa IP. El segundo hace el proceso inverso.

En el hilo de transmisión, inicialmente se realiza el control del flujo comprobando si ha pasado el tiempo IPG (Inter Packet Gap) requerido para poder enviar el paquete de datos (Jacobson, 1995), basado a su vez en el estado del bit BCN del último paquete recibido por el hilo de recepción (BCNr). Antes de enviar el paquete de datos a la capa IP, se encapsulan los datos con el número de secuencia, la identificación del protocolo y

finalmente la información de congestión (bit BCNs) localizada en el hilo de recepción mediante la comprobación de los números de secuencia recibidos.

En el hilo de recepción, inicialmente se comprueban los números de secuencia y se actualiza el bit BCN a enviar por el hilo de transmisión (bit BCNs). Tras ello se comprueba el bit BCN recibido (bit BCNr), que sirve para la actualización del IPG del control de flujo en transmisión, y se entregan los datos a la capa de aplicación.

### 3.1 Bidireccionalidad

El aprovechamiento de la bidireccionalidad se basa en la inserción de los datos de estados de red en el flujo de datos de teleoperación inverso, como puede apreciarse en la figura 2. Cuantitativamente, ello implica una disminución del caudal total necesario para el envío de la misma información, manteniéndose o mejorándose las características de estabilidad y pasividad del sistema teleoperado, al disminuir la carga sobre la red.

Si el caudal total utilizado en Internet por el flujo de maestro a esclavo se representa por  $C^{MS}$  y el caudal generado por el flujo esclavo a maestro por  $C^{SM}$ , se tiene

$$\begin{aligned} C^{MS} &= C_{data}^{MS} + C_{ack}^{MS} \\ C^{SM} &= C_{data}^{SM} + C_{ack}^{SM} \end{aligned} \quad (1)$$

siendo  $C_{data}$  y  $C_{ack}$ , los caudales generados por los datos de teleoperación y los asentamientos respectivamente, en cada uno de los sentidos de la teleoperación.

Partiendo de protocolos que generan un asentimiento ACK por cada paquete recibido, entonces

$$C_{data}|_{pqt/seg} = C_{ack}|_{pqt/seg} \quad (2)$$

de donde se deduce

$$C_{ack} = C_{data} * \frac{TA + TC}{TD + TC} \quad (3)$$

siendo  $TA$ ,  $TD$  y  $TC$  los tamaños de los campos de asentimiento, datos y cabecera respectivamente.

Si se representa  $C^{\uparrow}$  como el flujo de datos en uno de los sentidos y  $C^{\downarrow}$  el flujo total en sentido contrario generado por el tráfico de teleoperación, se puede decir que el tráfico total generado en Internet por la aplicación de teleoperación es

$$\begin{aligned} C_{total} &= C^{\uparrow} + C^{\downarrow} \\ C^{\uparrow} &= C_{data}^{MS} + C_{ack}^{MS} \\ C^{\downarrow} &= C_{data}^{SM} + C_{ack}^{SM} \end{aligned} \quad (4)$$

Suponiendo razonablemente que el formato de los datos y de los asentamientos es el mismo en ambos sentidos de la comunicación, de 4 y de 3 se deduce

$$C_{total} = (1 + \frac{TA + TC}{TD + TC}) * (C_{data}^{MS} + C_{data}^{SM}) \quad (5)$$

Si el asentimiento ACK no lleva datos de aplicación y sólo tiene la cabecera de transporte,  $TA = 0$ , la ecuación 5 queda de la forma

$$C_{total} = (1 + \frac{TC}{TD + TC})(C_{data}^{MS} + C_{data}^{SM}) \quad (6)$$

$$C_{total|NoBidirec} = (1 + \frac{TC}{TD + TC})C_{total|Bidirec}$$

La ecuación 6 cuantifica el aprovechamiento de la bidireccionalidad en el tráfico total de la aplicación de teleoperación supermedia en Internet, respecto a esquemas que no lo hacen, aumentando la cantidad de caudal disponible para el envío de la misma información y el mantenimiento de las mismas funcionalidades de transporte. Esto es, se eliminaría el caudal consumido por el esquema de control de protocolo (caudal útil a nivel de transporte), para aprovecharlo como caudal útil a nivel de aplicación. Se ha hecho una reproducción particular del esquema de (Lian *et al.*, 2002) en la figura 5 para mostrar los efectos de la bidireccionalidad. En la figura 5 se muestran tres tipos de sistemas de control y el efecto del muestreo o la velocidad de transmisión sobre los mismos. El control continuo no se ve afectado por diferentes velocidades de transmisión; el control digital tiene un rendimiento aceptable a partir de una velocidad de muestreo A; el control en red (aplicable a teleoperación) tiene un rendimiento aceptable a partir de una velocidad de transmisión B hasta una velocidad C. El punto C existe debido a los efectos negativos sobre el rendimiento que produciría una sobrecarga sobre la red. En el sistema de control en red con aprovechamiento bidireccional, los puntos B y C se trasladan a los puntos B' y C', aumentando el ancho de banda de transmisión aceptable en un factor  $(1 + \frac{TC}{TD+TC})$  dado por el resultado de la ecuación 6.

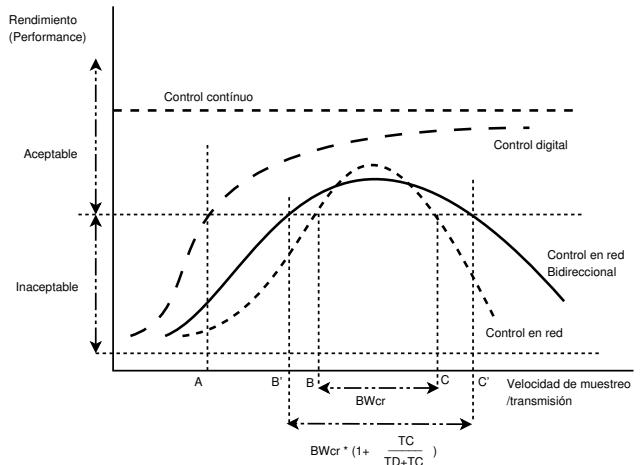


Figura 5. Comparación de rendimientos de control continuo, digital y en red con y sin aprovechamiento bidireccional.

En la figura 10 se comparan los protocolos UDP, TCP, TDTP y algoritmo Trinomial básico, desde el punto de vista del aprovechamiento de la bidireccionalidad, y teniendo en cuenta la eficiencia de los mismos.

### 3.2 Control del Flujo

El sistema de control del flujo, responde de forma parecida a los esquemas clásicos basados en emisor (*sender based*)

donde el receptor informa al emisor mediante paquetes ACK tipo asentimiento, del número de secuencia de los paquetes recibidos, detectando el emisor paquetes perdidos y valores RTT, ajustando sus parámetros de período o velocidad de envío en función de los resultados detectados.

La solución propuesta implementa el tipo de control trinomial (Liu *et al.*, 2002), (Liu *et al.*, 2003a), (Liu *et al.*, 2003b) y (Liu *et al.*, 2005), con algunas particularidades basadas en la bidireccionalidad del tráfico supermedia, como puede ser la creación del diagrama de estados de la figura 7. Este algoritmo de control de flujo ofrece una regulación de la velocidad de envío de datos en función de determinados estados de la red, manteniendo las características *TCPFriendly*, *Fairness* y *Best Effort*. Esto hace que ofrezca servicios de transporte más adecuados que TCP y menos canibalizadores que UDP. La comprobación de la característica *TCPFriendly* se realiza partiendo de igualar la velocidad de transmisión de un flujo TCP con la velocidad de un flujo del protocolo a estudiar. La ecuación  $\lambda_{max} = \lambda_{tcp} = \sqrt{3}/(RTT\sqrt{2p})$  resume ese punto de partida, donde  $\lambda_{tcp}$  es la velocidad de transmisión (en paquetes/segundo) de un flujo TCP,  $\lambda_{max}$  es la velocidad del flujo utilizando el protocolo que tiene que cumplir la condición *TCPFriendly* (Liu *et al.*, 2002) y  $p$  es la tasa de pérdidas.

El funcionamiento de este tipo de control de flujo tiene una base del tipo Equation-Based, donde la época de incremento de la velocidad de envío de datos (descenso de IPG) se produce por la ecuación (7) y el descenso de la velocidad de envío (incremento de IPG) se produce por la ecuación (8). Otros tipos de control del flujo como el TCP están basados en esquemas AIMD (Aditive Increment Multiplicative Decrement).

$$IPG_{i+1} = \frac{IPG_i \times W}{IPG_i + W} \quad (7)$$

$$IPG_{i+1} = \frac{IPG_i}{\beta} \quad (8)$$

En las ecuaciones anteriores tiene que cumplirse  $W > 0$  y  $\beta < 1$ .  $IPG_i$  es el tiempo entre envío de paquetes en la iteración  $i$ ,  $W$  es una variable dependiente de  $\alpha$  según la ecuación (9), que a su vez es dependiente de  $\beta$  y  $\gamma$  que en conjunto forman las variables del algoritmo trinomial.  $\beta$  y  $\gamma$  son constantes definidas al comienzo del protocolo, y  $\alpha$  varía para cada época de crecimiento según la ecuación (10).

$$W = RTT \left( \frac{\alpha}{i} \right)^\gamma \quad (9)$$

$$\alpha = K_1 S_0^{K_2} \quad (10)$$

$S_0$  es la última velocidad de transmisión (en paquetes por RTT segundos) antes de la época de incremento de la velocidad y está por tanto sujeto a la ecuación (11) donde  $IPG_0$  es el tiempo entre paquetes de comienzo de la época de crecimiento de la velocidad. Las constantes  $K_1$  y  $K_2$  dependen de  $\gamma$  y  $\beta$  según las ecuaciones (12) y (13) y por tanto son calculadas al inicio del protocolo. En la figura 6 pueden visualizarse los parámetros  $S_0$ ,  $i$  y  $RTT$  utilizados por el algoritmo.

$$S_0 = \frac{RTT}{IPG_0} \quad (11)$$

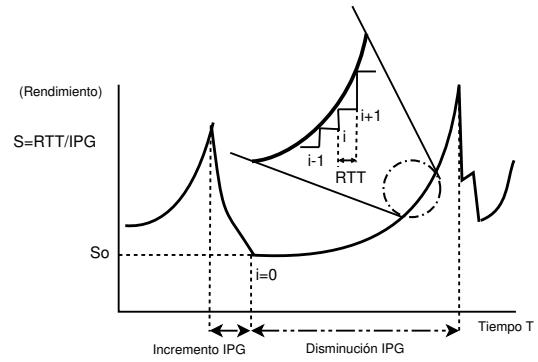


Figura 6. Variables de algoritmo trinomial.

$$K_1 = \frac{(\gamma + 1 - \gamma\beta)\Gamma(\gamma + 1)^{\frac{1}{\gamma}}}{1,5(\gamma + 1)\beta^{\frac{1}{\gamma}}(1 - \beta)^{\frac{(\gamma - 1)}{\gamma}}} \quad (12)$$

$$K_2 = \frac{\gamma - 1}{\gamma} \quad (13)$$

A efectos prácticos, en la fase de crecimiento de la frecuencia de envío de paquete, el algoritmo permite cambios de crecimiento con la variable  $W$ , de forma que si  $W$  se hace muy grande,  $IPG_{i+1} \simeq IPG_i$  y por tanto el crecimiento es prácticamente nulo y la velocidad de transmisión de paquete tiene un crecimiento bajo; esto sucede al comienzo del periodo de crecimiento, cuando las iteraciones  $i$  son pocas. El crecimiento de las iteraciones implica que  $W$  tienda a 0, y por tanto  $IPG_{i+1}$  tienda a cero, que viene a decir que el crecimiento de la frecuencia de envío es muy alto. En la fase de decrecimiento, la iteración es más sencilla, limitándose a bajar la frecuencia de envío en función de la variable  $\beta$ . El parámetro  $\beta$  define el aumento del  $IPG$  cuando se detecta pérdida de paquetes y el parámetro  $\gamma$  determina la velocidad con la que coge ancho de banda disponible cuando no hay pérdida de paquetes.

El control de flujo trinomial implementado en TDTP responde al diagrama de estados de la figura 7. Este diagrama dispone de cuatro estados como se puede ver en el Cuadro 1, dependiendo del estado anterior y actual del bit  $BCN_r$ . En el diagrama se han adoptado decisiones sobre los momentos y frecuencia de actualización de los parámetros  $RTT$ ,  $i$  y  $\alpha$ . El parámetro  $\alpha$  se actualiza al principio de cada fase de crecimiento (disminución  $IPG$ ) según especificación del trabajo original (Liu *et al.*, 2003a). El parámetro  $RTT$  se actualiza igualmente al principio de cada fase de crecimiento con el último valor calculado del paquete recibido, en lugar de hacerlo tras la llegada de cada paquete de retorno (bit  $BCN_r$ ) y el parámetro contador  $i$  es el único que hace que  $W$  decrezca en la fase de crecimiento cada  $RTT$  segundos. El trabajo original de presentación del algoritmo trinomial no limita en la forma en que se adquieren los valores de sus parámetros, dejándose esa decisión a los implementadores. En este trabajo se ha buscado la minimización de las necesidades de cómputo manteniendo resultados más que razonables.

Tabla 1. Estados de algoritmo de Control de flujo

Estado	Estado anterior	bit BCN recibido
IPG decrece	IPG decrece	0
IPG pasa a crecer	IPG decrece	1
IPG crece	IPG crece	1
IPG pasa a decrecer	IPG crece	0

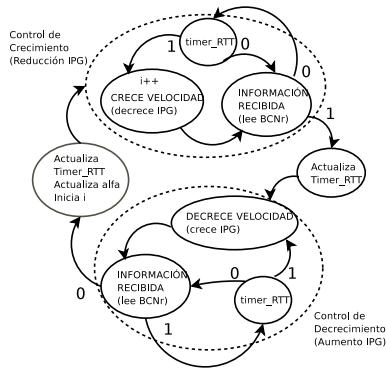


Figura 7. Diagrama de estados del control de flujo.

### 3.3 Compatibilidad con ACK

La compatibilidad del protocolo con tipos de tráfico que sí disponen de protocolos específicos como el vídeo y el audio en tiempo real (RTP) se ofrece como la posibilidad de adecuar los “datos multimedia” a los flujos supermedia ofrecidos por el TDTP. Básicamente se tendría que encargar la capa de aplicación de dividir los paquetes en tamaños fijos y reensamblarlos en destino. En el caso en que el tráfico multimedia sea el único retorno del sistema de teleoperación y que escoja otros protocolos y aplicaciones para transmitirse, TDTP proporciona un mecanismo basado en el clásico envío de ACK’s de retorno para el control del flujo en origen, controlados mediante temporizadores. Este mecanismo sirve también para el caso en que uno de los flujos de datos se vea cortado o sufra pérdidas suficientemente amplias como para que se considere inexistente. La figura 8 presenta un diagrama de estados de este mecanismo.

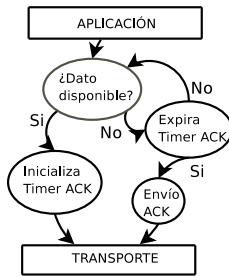


Figura 8. Mecanismo de ACK

## 4. PAQUETE DE DATOS

El paquete de datos se enfoca principalmente desde el formato de la cabecera y su relación con el campo de datos en lo que a tamaño se refiere. Se particulariza esa relación para el caso de datos de teleoperación con tamaños bastante pequeños. Se presenta el formato de la cabecera de transporte del protocolo, y se detallan los campos, incluidos aquellos aprovechables por la cabecera de red IPv4.

### 4.1 Tamaño y Eficiencia

El tamaño del paquete es responsable junto con el control del flujo de la tasa de transmisión del canal de datos de teleoperación. Un paquete básico de datos utilizando el protocolo IP/UDP tiene 28 bytes de cabecera. Una estructura de datos común en sistemas teleoperados de  $n$  grados de libertad puede ser la de disponer de 1 octeto de control, 2 octetos de datos por

cada grado de libertad y finalmente 2 octetos de información auxiliar, controlada por el byte de control (Emanuel Slawinski and Mut, 2007). Esto sumaría 9 octetos de datos en total en el caso de 3 grados de libertad. La resolución de datos planteada corresponde a un tipo de dato *short int* de 2 octetos, en principio suficiente para muchos sistemas. Si en lugar de tipo de datos *short int* de 2 octetos, se necesitan resoluciones mayores, se le añaden los datos básicos de teleoperación de un sistema de tres grados de libertad de movimiento de alta resolución, conformando estos 12 octetos más, 4 octetos por cada grado de libertad, es decir, un dato de tipo *long int* con rango que va desde -2,147,483,648 hasta +2,147,483,647. En ese caso el total de datos útiles de teleoperación en un sentido serían de 15 octetos, aún así menor que los 28 octetos que conformaría la cabecera de Internet mediante el transporte UDP.

A nivel de eficiencia ( $Tamaño_{datos}/Tamaño_{cabecera}$ ), el paquete básico de teleoperación sobre IP/UDP es poco eficiente. La eficiencia mejora a mayor resolución de movimiento o cantidad de datos a transmitir, aunque también aumenta en ese caso el caudal requerido, y en muchos casos no es necesario. Aunque se decida por estructuras de datos diferentes, el tamaño de los mismos será parecido en ambos sentidos de transmisión. El conocimiento del tamaño de los datos puede hacerse mediante dos procedimientos:

- Tamaño de datos fijo, asignado al principio de la transmisión mediante un protocolo de anuncio y parametrización.
- Utilización del campo “Longitud total del paquete” de la cabecera IPv4, al que se le pueden restar el tamaño de la cabecera de transporte al ser este fijo.

En un protocolo de datos de teleoperación haptica o basada en sensores/actuadores, un tamaño de cabecera pequeño puede mejorar muy ostensiblemente la eficiencia de la transmisión. La cabecera del paquete de datos propuesto aprovecha una buena parte de campos de la cabecera IPv4 para los servicios de identificación y secuenciación del tráfico, y crea cuatro campos nuevos específicos del transporte, mas un campo de control de errores, los campos BCN y ACK de un bit de longitud, el campo opciones de 6 bits de longitud, el campo Seq\_ACK, de dos octetos de longitud y el campo de error, de dos octetos. El formato de la cabecera queda representado en la figura 9.

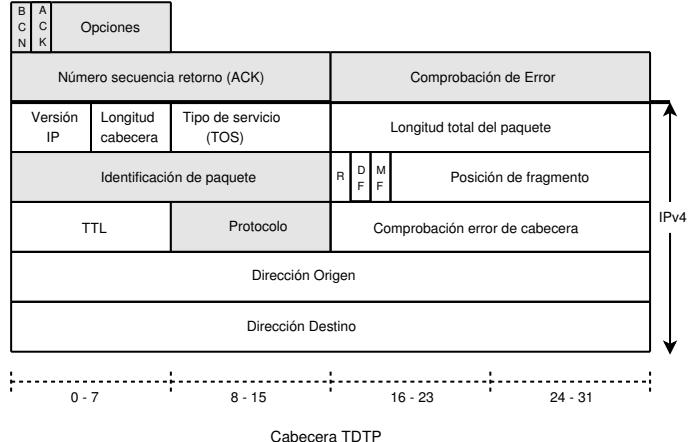


Figura 9. Cabecera TDTP

La mejora de la eficiencia repercute directamente en un menor requerimiento de caudal para el envío de la misma cantidad de datos. La fórmula 14 representa dentro de una arquitectura

concreta, el caudal de una capa  $k$  conocido el caudal de una capa  $m$ , en condiciones ideales de transmisión continua sin retransmisiones ni errores.  $C_k$  es el caudal en la capa  $k$  y  $TP_k$  es el tamaño del paquete (datos + cabeceras) en la capa  $k$ .

$$C_k = C_m * \frac{TP_k}{TP_m} \quad (14)$$

Si se aplica la ecuación 14 a los caudales que se entregan a Internet y que son generados por un operador maestro o los sensores del esclavo, el tamaño del paquete cuando es entregado a las capas de enlace subyacentes a IP tendrá como añadido a los datos, las cabeceras de transporte e IP. La ecuación 15 representa ese caso, donde  $TD$  es el tamaño de los datos y  $TC$  es el tamaño de la cabecera.

$$C_{Internet} = C_{aplicación} * \frac{TD + TC}{TD} \quad (15)$$

De la ecuación 15 se deduce que a mayor tamaño de datos, menos variable es el caudal entregado a Internet respecto al caudal generado por la aplicación de teleoperación y entregado a la capa de transporte. En cambio, a menor tamaño de los datos generados en la aplicación, más influencia tiene el tamaño de la cabecera en el aumento del caudal entregado a la red respecto al caudal generado.

Si se combina la ecuación 6 donde se cuantifica la mejora de la bidireccionalidad, con la ecuación 15 donde se cuantifica la eficiencia del formato del paquete de datos de teleoperación en Internet, se tiene para un enlace extremo a extremo simétrico donde el caudal de aplicación maestro-esclavo es igual al caudal de aplicación esclavo-maestro:

$$C_{total} = 2 * C_{aplicación} * \left( \frac{TD + 2TC}{TD} \right) \quad (16)$$

para el caso en que no existe aprovechamiento de la bidireccionalidad (como TCP, TFRC, ETP entre otros) y con los asentimientos ACK embebidos en la cabecera de transporte, y la ecuación

$$C_{total} = 2 * C_{aplicación} * \left( \frac{TD + TC}{TD} \right) \quad (17)$$

para el caso en que existe un aprovechamiento de la bidireccionalidad como en el protocolo TDTP.

La figura 10 presenta de forma gráfica la comparación entre TCP, UDP, TDTP y Trinomial puro. En este caso, el protocolo UDP se presenta como protocolo que aprovecha la bidireccionalidad, pues no tiene tráfico añadido de asentimientos.

#### 4.2 Direcciónamiento e Identificación

La cabecera IPv4 queda definida en el RFC 791 (Postel, 1981). La cabecera IPv4 es utilizada como direcciónamiento e identificación del flujo TDTP, o incluso para conocer el tamaño del paquete de datos. La cabecera IP proporciona direcciónamiento extremo a extremo mediante los campos de direcciones IP origen y destino, y cada uno de los extremos tiene que discriminar diferentes tipos de tráfico en la capa de transporte. La discriminación se realiza mediante el campo “Protocol” de la cabecera IP, de forma que ese campo se convierte en uno mas

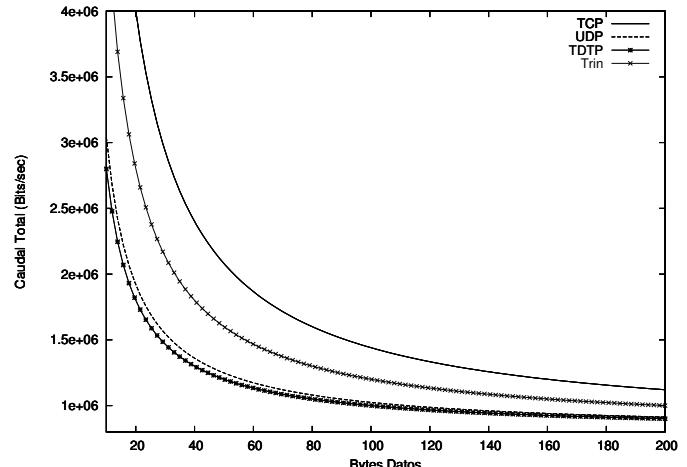


Figura 10. Bireccionalidad TCP, UDP, TDTP y Trinomial

de la capa de transporte propuesta. Así, los procesos maestro y esclavo pueden capturar los datos que les llegan sin disponer de un sistema de direccionamiento interno como lo son los puertos UDP. El IANA dispone de los identificadores del campo “Protocol” públicos en Internet, estando a fecha actual reservados para experimentación y testeo los identificadores 253 y 254 (Arkko and Bradner, 2008).

El direccionamiento de actuadores y sensores debe ser implementado en la estructura de datos del protocolo, y por tanto negociado previamente. Esta negociación puede realizarse mediante protocolo de anuncio, o mediante la fijación previa de los parámetros por parte de los operadores.

#### 4.3 Campo Opciones

Este campo está dividido en dos indicadores de un bit y 6 bits para opciones futuras u otros sistemas de control del flujo.

**Bit BCN** Es el bit que indica al extremo opuesto si se ha producido o no una pérdida de paquete de los que ha recibido. Es el dato que se usa en recepción como criterio para el control del flujo. En transmisión, este bit se pone a 1 cuando detecta un fallo de secuencia en los paquetes entrantes, o cuando expira un temporizador que tiene que ser mayor que el temporizador ACK indicado a continuación, pues básicamente indica que hay un hueco en la recepción y lo interpreta como una pérdida de paquetes.

**Bit ACK** Si este bit está puesto a 1, es que el paquete sólo lleva información del estado de la red sin datos de teleoperación. Se pone a 1 cuando expira un temporizador que indica el tiempo sin que se estén generando datos de teleoperación de la capa de aplicación. Básicamente mantiene el sistema de control del flujo aunque no existan datos en el sentido opuesto. Si ACK está a 0, la información del estado de la red de la cabecera va acompañada por un campo de datos de teleoperación, proporcionados por la capa de aplicación. Cuando las aplicaciones de ambos extremos no tienen datos que transmitir, el flujo de paquetes ACK mantiene el canal de comunicación abierto.

**Opciones Futuras** Estos 6 bits están reservados para opciones futuras o el envío de mas información de control del protocolo, como podría ser la velocidad a la que tendría que transmitir el destino, habitual en algún otro protocolo como el TEAR (Rhee

et al., 2000). La posibilidad de escoger otro sistema de control del flujo mantiene ese campo con bastantes opciones de ser utilizado.

#### 4.4 Campo Seq-ACK

Este campo retorna el número de secuencia de un paquete recibido a su origen. El tiempo de llegada será comparado con el de salida, almacenado en una tabla, para disponer de una estimación del RTT. El tamaño de este campo es de 2 octetos, para mantener la compatibilidad con el campo Identificador de la cabecera IP.

#### 4.5 Campo Seq

Este campo es realmente el campo “Identificador de paquete” de la cabecera IP, reconvertido como número de secuencia del protocolo de transporte STDP. Tiene 2 octetos de longitud.

La cabecera IP utiliza este campo para distinguir los fragmentos de un datagrama de los de otro en el caso en que se produzca fragmentación. El valor de Identificación es puesto por el origen del datagrama a un valor único para ese protocolo y por origen-destino mientras circula por Internet. Para un datagrama formado por un sólo paquete, se pone el indicador de “mas fragmentos” a cero (campo de un bit MF de la cabecera IPv4) y el Fragment-Offset (campo “Posición de Fragmento” de la cabecera IPv4) también a cero, y sólo habrá un paquete de datos en Internet que lleve ese identificador y el par origen-destino. El campo Identificador es asignado por el emisor y no es modificable por los routers intermedios.

#### 4.6 Otros servicios de la cabecera IP

La cabecera IPv4 dispone de un campo de un octeto de longitud llamado “Tipo de Servicio” o “TOS”, que proporciona la posibilidad de establecer prioridades y algunos tipos de servicio sobre los datagramas. El uso de estos servicios no siempre está claro, e incluso en algunos casos desaconsejado, pues el uso de demasiados de esos parámetros puede perjudicar el rendimiento.

### 5. SIMULACIÓN

Las simulaciones se han realizado con el entorno abierto NS2 (*Network Simulator 2*). Para ello se ha implementado la algoritma adecuada al protocolo propuesto y se han realizado comparaciones con otros protocolos. Las topologías mas utilizadas (Floyd and Fall, 1999), (Rejaie et al., 1999), (Liu et al., 2003a), (Zhou et al., 2006) para la simulación pueden representarse en la figura 11, y tienen características suficientemente flexibles para estudiar determinadas situaciones de ocupación de enlaces o de inclusión de tipos de tráfico. Al trabajarse sobre capa de transporte, las repercusiones que tendrían diferentes tecnologías de acceso quedarían contempladas en las topologías planteadas, pudiéndose extender hacia topologías inalámbricas, también ampliamente soportadas en NS2.

La topología utiliza un enlace troncal entre R1 y R2 que sirve de cuello de botella para conocer las prestaciones del esquema en situaciones de congestión. La capacidad de los enlaces de acceso hacia los nodos 1 a 20, maestro y esclavo son de 100Mbps, y la capacidad del troncal es de 1Mbps. Todos los enlaces son simétricos.

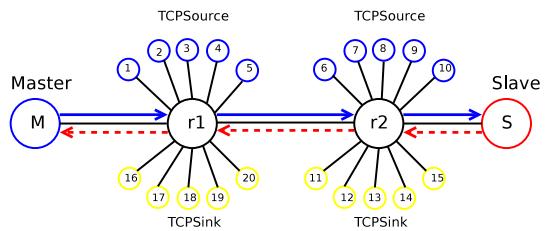


Figura 11. Escenario de simulación NS2.

Las simulaciones se han realizado a lo largo de un total de 100 segundos, manteniendo tráfico de tipo TCP en los nodos del 1 al 20, de forma que los nodos 1-5 envían tráfico FTP a los nodos 11-15, y los nodos 6-10 envían tráfico FTP a los nodos 16-20. El enlace entre R1 y R2 está inicialmente ocupado, para dejarse libre a los 20 segundos. Progresivamente vuelve a ocuparse a partir de los 30 segundos, para poco a poco dejarse de nuevo totalmente libre a los 80 segundos. En el sentido R2-R1 se mantiene ocupado desde los 10 segundos a lo largo de todo el tiempo de simulación.

La característica *TCPFriendly* de cada uno de los protocolos puede quedar reflejada en una ocupación media del enlace que hace de cuello de botella en torno a la sexta parte de la capacidad disponible, en el periodo en que el enlace está ocupado, ya que existen otros cinco flujos TCP ocupando el enlace. La característica best-effort puede reflejarse en la rapidez de cambio a la disponibilidad de mas recursos en el canal.

Las gráficas 12 y 13 muestran los resultados de rendimiento y porcentaje de pérdidas de los protocolos TDTP, Trinomial puro, TFRC y TEAR en las direcciones maestro → esclavo y esclavo → maestro. Para los protocolos TDTP y Trinomial pueden apreciarse respuestas bastante rápidas a los cambios en la disponibilidad de recursos, premiándose en cambio la continuidad en los protocolos TFRC y TEAR.

En las figuras pueden apreciarse varias conclusiones:

- Los protocolos TDTP y Trinomial, con esquemas de control del flujo iguales, tienen una notable diferencia en el rendimiento al aprovecharse el protocolo TDTP de la bidireccionalidad. El aumento de las pérdidas es prácticamente inapreciable, estando en órdenes entre el 0 y el 1 por cien.
- En el protocolo TFRC se ha utilizado la variante para tráfico de tipo VoIP. Puede verse continuidad y lenta adaptación a la disponibilidad. En este caso, al utilizar la variante para VoIP y por tanto para paquetes del orden de los 250 octetos, está algo mas adaptado para tráfico de teleoperación.
- En el protocolo TEAR puede apreciarse un escaso rendimiento para paquetes de tamaño pequeño, al ser la tasa de crecimiento y decrecimiento de transmisión muy moderada para garantizar estabilidad de la transmisión. El protocolo TEAR muestra de forma notable mejores resultados al utilizarse paquetes de tamaños superiores a los 500 octetos.

### 6. CONCLUSIONES Y TRABAJO FUTURO

El esquema de protocolo propuesto cumple con especificaciones exclusivas de sistemas de teleoperación, como la bilateridad de la transmisión, los requisitos de tiempo real y la estructura de datos de teleoperación de tipo háptico u otros que

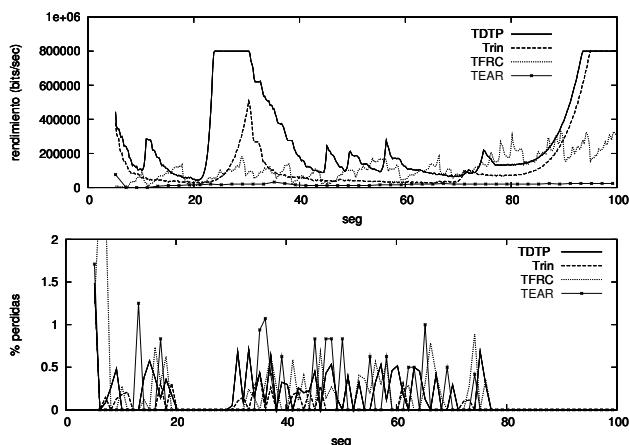


Figura 12. Flujo de datos de maestro a esclavo. Rendimiento y % de pérdidas

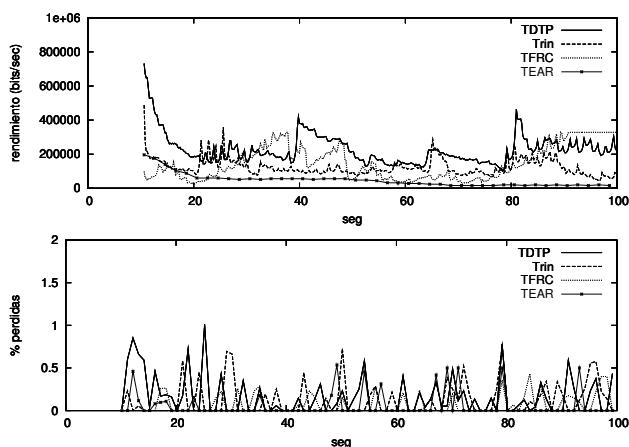


Figura 13. Fluxos de dados de esclavo a maestro. Rendimiento y % de pérdidas

conlleven un tamaño pequeño de datos. El sistema de control de flujo propuesto es el algoritmo trinomial, lo que hace que cumpla con las características de *TCPFriendly* y *Fairness* necesarias para que el protocolo sea aceptado en la red pública Internet con las mismas características que los protocolos de transporte habituales. El escenario de tráfico propuesto no envía un ACK por cada paquete transmitido o recibido, sino que establece un flujo de datos bidireccional que incluye a los ACK's del sentido contrario, creándose unas correlaciones que están abiertas a nuevos análisis. Los resultados de las simulaciones muestran un mejor aprovechamiento de los recursos disponibles, mejorando el rendimiento con tasas de pérdidas similares.

Como línea futura se plantea el estudio e implementación de otros modelos de control del flujo, en particular el TEAR y las variantes TFRC o DCCP. En el caso del protocolo TEAR, este protocolo mantiene una cierta independencia entre la tasa de pérdidas y la respuesta inmediata en la velocidad de transmisión de los datos, al no realizarse asentamientos ACK por cada paquete recibido ni de forma regulada en sistemas de ventanas. En el caso de las variantes TFRC, TFRC-SP, DCCP, resulta muy interesante el esquema de utilización de diferentes opciones de control del flujo propuestas por el DCCP, en CCID2, CCID3 y CCID4.

## AGRADECIMIENTOS

Este trabajo ha sido realizado parcialmente gracias al apoyo del Ministerio de Ciencia y Tecnología de España en el marco de los proyectos CICYT DPI2007-66455-C02-02 y DPI2008-06738-C02-03. Además ha recibido el apoyo de la Consellería de Innovación e Industria de la Xunta de Galicia en el mismo país, en el marco del Proyecto 08DPI-011303PR.

## REFERENCIAS

- Andersen, David, Hari Balakrishnan, Frans Kaashoek and Robert Morris (2001). Resilient overlay networks. *SIGOPS Oper. Syst. Rev.* **35**(5), 131–145.
- Anderson, R.J. and M.W. Spong (1988). Bilateral control of teleoperators with time delay. *Systems, Man, and Cybernetics, 1988. Proceedings of the 1988 IEEE International Conference on* **1**, 131–138.
- Arkko, J. and S. Bradner (2008). IANA Allocation Guidelines for the Protocol Field. RFC 5237 (Best Current Practice).
- Berestesky, P., N. Chopra and M.W. Spong (2004). Discrete time passivity in bilateral teleoperation over the internet. *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on* **5**, 4557–4564 Vol.5.
- Cen, Zhiwei, Matt W. Mutka and Danyu Zhu (2005). An overlay network transport service for teleoperation systems. Technical report.
- Decotignie, J.-D. (2005). Ethernet-based real-time and industrial communications. *Proceedings of the IEEE* **93**(6), 1102–1117.
- Diaz-Cacho, M., A. Fernandez and A. Barreiro (2008). Sistema de teleoperación colaborativa grúa-cámaras con retorno de estado. Website. <http://www.cea-ifac.es/actividades/jornadas/XXIX/pdf/271.pdf>.
- Duan, Zhenhai, Zhi-Li Zhang and Y.T. Hou (2003). Service overlay networks: Slas, qos, and bandwidth provisioning. *Networking, IEEE/ACM Transactions on* **11**(6), 870–883.
- Emanuel Slawinski, Jose F. Postigo and Vicente Mut (2007). Bilateral teleoperation through the internet. *Robot. Auton. Syst.* **55**(3), 205–215.
- Emanuel Slawinski, José F. Postigo y Vicente Mut (2006). Experiencias en teleoperación bilateral de robots. *Revista iberoamericana de automática e informática industrial (RIAI)* **3**(1), 29–38.
- Felser, M. (2005). Real time ethernet. industry prospective. *Proceedings of the IEEE* **93**(6), 1118–1129.
- Fernandez Villaverde, A., C. Raimundez Alvarez and A. Barreiro Bias (2007). Digital passive teleoperation of a gantry crane. *Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on* pp. 56–61.
- Floyd, S. and E. Kohler (2007). TCP Friendly Rate Control (TFRC): The Small-Packet (SP) Variant. RFC 4828 (Experimental).
- Floyd, S. and J. Kempf (2004). IAB Concerns Regarding Congestion Control for Voice Traffic in the Internet. RFC 3714 (Informational).
- Floyd, S. and K. Fall (1999). Promoting the use of end-to-end congestion control in the internet. *Networking, IEEE/ACM Transactions on* **7**(4), 458–472.
- Floyd, Sally, Mark Handley, Jitendra Padhye and Jörg Widmer (2000). Equation-based congestion control for unicast applications. *SIGCOMM Comput. Commun. Rev.* **30**(4), 43–56.

- García-Rivera, Matías and Antonio Barreiro (2007). Analysis of networked control systems with drops and variable delays. *Automatica* **43**(12), 2054–2059.
- Group, Audio-Video Transport Working, H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson (1996). RTP: A Transport Protocol for Real-Time Applications. RFC 1889 (Proposed Standard). Obsoleted by RFC 3550.
- Handley, M., S. Floyd, J. Padhye and J. Widmer (2003). TCP Friendly Rate Control (TFRC): Protocol Specification. RFC 3448 (Proposed Standard). Obsoleted by RFC 5348.
- Hirche, S., P. Hinterseer, E. Steinbach and M. Buss (2005). Towards deadband control in networked teleoperation systems. In: *In: Proceedings of the 16.th IFAC World*.
- Jacobson, V. (1995). Congestion avoidance and control. *SIGCOMM Comput. Commun. Rev.* **25**(1), 157–187.
- Jannotti, John, David K. Gifford, Kirk L. Johnson, M. Frans Kaashoek and Jr. James W. O'Toole (2000). Overcast: reliable multicasting with on overlay network. In: *OSDI'00: Proceedings of the 4th conference on Symposium on Operating System Design & Implementation*. USENIX Association. Berkeley, CA, USA. pp. 14–14.
- Kohler, E., M. Handley and S. Floyd (2006a). Datagram Congestion Control Protocol (DCCP). RFC 4340 (Proposed Standard).
- Kohler, Eddie, Mark Handley and Sally Floyd (2006b). Designing dccp: congestion control without reliability. In: *SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM. New York, NY, USA. pp. 27–38.
- Kopetz, Hermann (1997). *Real-Time Systems : Design Principles for Distributed Embedded Applications (The International Series in Engineering and Computer Science)*. Springer.
- Li, Zhi and P. Mohapatra (2004). Qron: Qos-aware routing in overlay networks. *Selected Areas in Communications, IEEE Journal on* **22**(1), 29–40.
- Lian, Feng-Li, J. Moyne and D. Tilbury (2002). Network design consideration for distributed control systems. *Control Systems Technology, IEEE Transactions on* **10**(2), 297–307.
- Liu, Peter Xiaoping, Max Q.-H. Meng and Simon X. Yang (2003a). Data communications for internet robots. *Auton. Robots* **15**(3), 213–223.
- Liu, Peter Xiaoping, M.Q.-H. Meng, Jason Gu, S.X. Yang and Chao Hu (2003b). Control and data transmission for internet robots. *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on* **2**, 1659–1664 vol.2.
- Liu, P.X., M. Meng, Xiufen Ye and J. Gu (2002). An udp-based protocol for internet robots. *Intelligent Control and Automation, 2002. Proceedings of the 4th World Congress on* **1**, 59–65 vol.1.
- Liu, P.X., M.Q.-H. Meng, P.R. Liu and S.X. Yang (2005). An end-to-end transmission architecture for the remote control of robots over ip networks. *Mechatronics, IEEE/ASME Transactions on* **10**(5), 560–570.
- Munir, S. and W.J. Book (2001). Internet based teleoperation using wave variables with prediction. *Advanced Intelligent Mechatronics, 2001. Proceedings. 2001 IEEE/ASME International Conference on* **1**, 43–50 vol.1.
- Niemeyer, G. and J.-J.E. Slotline (1991). Stable adaptive teleoperation. *Oceanic Engineering, IEEE Journal of* **16**(1), 152–162.
- Ping, Li, Lu Wenjuan and Sun Zengqi (2005). Transport layer protocol reconfiguration for network-based robot control system. *Networking, Sensing and Control, 2005. Proceedings.* 2005 IEEE pp. 1049–1053.
- Postel, J. (1981). Internet Protocol. RFC 791 (Standard). Updated by RFC 1349.
- Rejaie, R., M. Handley and D. Estrin (1999). Rap: An end-to-end rate-based congestion control mechanism for realtime streams in the internet. *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* **3**, 1337–1345 vol.3.
- Rhee, Injong, Volkan Ozdemir and Yung Yi (2000). Tear: Tcp emulation at receivers - flow control for multimedia streaming. Technical report.
- Schiffer, V. (2001). The cip family of fieldbus protocols and its newest member ethernet/ip. *Emerging Technologies and Factory Automation, 2001. Proceedings. 2001 8th IEEE International Conference on* pp. 377–384 vol.1.
- Schulzrinne, H., S. Casner, R. Frederick and V. Jacobson (2003). RTP: A Transport Protocol for Real-Time Applications. RFC 3550 (Standard).
- Subramanian, Lakshminarayanan, Ion Stoica, Hari Balakrishnan and Randy H. Katz (2004). Overqos: an overlay based architecture for enhancing internet qos. In: *NSDI'04: Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation*. USENIX Association. Berkeley, CA, USA. pp. 6–6.
- Tipsuwan, Y. and Mo-Yuen Chow (2004). Gain scheduler middleware: a methodology to enable existing controllers for networked control and teleoperation - part i: networked control. *Industrial Electronics, IEEE Transactions on* **51**(6), 1218–1227.
- Uchimura, Y. and T. Yakoh (2004). Bilateral robot system on the real-time network structure. *Industrial Electronics, IEEE Transactions on* **51**(5), 940–946.
- Wirz, R., R. Marin, J.M. Claver, J. Fernandez and E. Cervera (2007). Transport protocols for remote programming of network robots within the context of telelaboratories for education: A comparative analysis. *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on* pp. 1315–1320.
- Wirz, Raul, Manuel Ferre, Raúl Marín, Jorge Barrio, José M. Claver and Javier Ortego (2008a). Efficient transport protocol for networked haptics applications. In: *EuroHaptics*. pp. 3–12.
- Wirz, Raul, Raul Marín, José M. Claver, Manuel Ferre, Rafael Aracil and Josep Fernández (2008b). End-to-end congestion control protocols for remote programming of robots, using heterogeneous networks: A comparative analysis. *Robotics and Autonomous Systems* **56**(10), 865 – 874. Network Robot Systems.
- Yutaka Uchimura, Nobuyuki Yamasaki and Kouhei Ohnishi (2005). Prioritized data transfer for a bilateral robot control via real-time network system. *IEEJ Transactions on Industry Applications* **125**(2), 199–204.
- Zhou, Yajin, M.Q.-H. Meng, Huawei Liang, Lei Sun, Zhi Xu and Kun Shen (2006). Tfrc-probe: A transport protocol for teleoperation systems of mobile robots. *Information Acquisition, 2006 IEEE International Conference on* pp. 1492–1496.