Multimedia Multi-Networking: a New Concept
Multi-réseaux Multimédia : un Nouveau Concept

Pascal Berthou, Thierry Gayraud,
Philippe Owezarski, and Michel Diaz

LAAS-CNRS
7, Avenue du Colonel Roche
31077, Toulouse cedex 4, France
e-mail: {berthou, gayraud, owe, diaz}@laas.fr

Abstract. Recent researches in communication systems are leading to the multiplication of communication technologies. Because of this trend there is now a very wide range of different kinds of networks from copper lines for telephony to high speed fibres, as well as satellite or wireless mobile networks. It would then be very useful to be capable of using all these new communication networks all together. We call this domain related to the use by an application of several different networks “multi-networking”. But the problematic of multi-networking is two folds: (1) First, it can be really interesting to have several network access and to be able to use them in parallel. For instance, it can consist, in the case of digital and interactive TV, of using digital satellite channels for broadcasting audio and video, and using the wire Internet or an ISDN (Integrated Services Digital Network) network to send specific data to dedicated users. This is what we call “parallel multi-networking”. (2) The second folder of this problematic deals with guaranteeing Quality of Service (QoS) while connections cross several networks or domains, especially when there are firewalls or NAT (Network Address Translation) servers in between that break the end to end IP model. In addition, the introduction of wireless or satellite links that have high delays and loss ratio inside the Internet can lead to important QoS degradation as wire Internet protocols are not efficient on wireless and satellite links. This aspect of the problematic is called “serial multi-networking”. This second aspect has been much more addressed in a recent past than the first one. It leads to some specific solutions, most of the time application oriented, as caching for web application for instance. To cope with other problems, as the introduction of satellite links in the Internet, proxies system have been designed to handle data flows before entering the satellite link. Proxies are, there, in charge of performing some spoofing operations. But in any case, even if there are some application specific solutions, or some network dedicated approaches, none of them is able to handle live real time traffic.

Hence, this work aims to propose a new solution relying on new protocols and architecture to cope with multi-networking. The solution for parallel multi-networking is called MMPOC-MN (MultiMedia Partial Order Connection for Multi-Networking). It is based on a partially ordered and reliable communication principle that allows us to reduce the end to end delay, and to enforce synchronisation between parallel flows on separated networks. This protocol can then be tuned very precisely in order to be the optimal transmission protocol according to application requirements and network constraints. To cope also with serial multi-networking, this protocol architecture has been extended. The new general (parallel and serial) multi-networking protocol is called MNP (Multi-Network Protocol). It is based on the concept of splitting the end to end connection in several trunks, each trunk being supported by a single network domain, each domain being supported by a single technology. Then, the best suited transmission parameters are used on each trunk, and the most suited spoofing algorithms are applied on data streams depending on the application requirement model. These protocols and architecture have been developed using the OPNET modeller and simulated, to evaluate the benefits of our solution. In this paper we are also focusing on how deploying such applicative protocol and architecture. The recommended solution consists in using active networking as ANTS, capable to download and run portable code on network components.

Résumé. Les travaux de recherche récents dans le domaine des systèmes de communication conduisent à une multiplication des technologies de communication. A cause de cette tendance, il existe aujourd’hui un très large choix de technologies réseaux différentes allant des lignes de

cuivre pour la téléphonie jusqu'à des fibres optiques haute vitesse, en passant par des réseaux mobiles sans fil ou satellites. Il serait donc extrêmement intéressant de pouvoir utiliser, tous ensembles, tous ces réseaux de communication. Ce domaine d'étude, consistant pour une application à utiliser plusieurs types de réseaux différents, s'appelle le « multi-réseau ». La problématique du multi-réseau comporte deux volets : (1) Premièrement, il peut être très intéressant d'avoir à sa disposition plusieurs accès réseaux et de pouvoir les utiliser simultanément / en parallèle. Par exemple, cela peut consister, dans le cas de la télévision numérique et interactive, à utiliser un canal satellite numérique pour diffuser l'audio et la vidéo et utiliser l'Internet filaire classique ou un réseau RNIS pour transmettre des données spécifiques à chaque utilisateur. C'est ce que nous appelons le « multi-réseau parallèle ». (2) Le second volet de cette problématique s'intéresse à comment garantir la qualité de service (Qo)S lorsque les connexions traversent plusieurs réseaux ou domaines de l'Internet, et en particulier lorsqu'il existe sur le chemin des serveurs NAT (Network Address Translation) ou des firewalls qui brisent le modèle IP de bout en bout. De plus, l'introduction de réseaux sans fil ou de liens satellites qui ont des délais et des taux de pertes importants peuvent conduire à des dégradations importantes de la QoS, car les protocoles de l'Internet ne sont pas du tout efficaces sur de tels liens. Cet aspect de la problématique est appelé « multi-réseau série ». Ce second aspect a été bien plus traité dans un passé récent que le premier. Cela a conduit à quelques solutions dédiées la plupart du temps à des applications spécifiques, comme le « *caching* » pour des applications du Web par exemple. Pour régler d'autres problèmes, comme l'introduction de liens satellites dans l'Internet, des systèmes de proxies ont été conçus pour prendre en charge les flux de données avant qu'ils n'entrent dans le réseau satellite. Les *proxies* sont, ici, responsables de la réalisation d'opérations de « *spoofing* ». Mais dans tous les cas, même s'il existe des solutions spécifiques pour certaines applications, ou des approches dédiées pour certains types de réseaux, aucune d'entre elles n'est capable de prendre en charge du trafic temps réel.

Par conséquent, ces travaux proposent une nouvelle solution, basée sur de nouveaux protocoles et de nouvelles architectures, pour gérer ce problème du multi-réseau. La solution pour le multi-réseau parallèle est appelée MMPOC-MN (MultiMedia Partial Order Connection for Multi-Networking). Elle est basée sur un principe de communication mettant en œuvre un ordre et une fiabilité partiels qui permettent de réduire le délai de bout en bout et de mettre en œuvre des mécanismes de synchronisation entre des flux parallèles transitant sur des réseaux séparés. Ce protocole peut être réglé de façon très précise pour optimiser les transmissions en fonction des besoins de l'application et des contraintes réseaux. Pour prendre en compte le multi-réseau série, cette architecture protocolaire a été étendue. La nouveau protocole général (parallèle et série) s'appelle MNP (Multi-Network Protocol). Il repose sur le concept de coupure de la connexion de bout en bout en plusieurs morceaux, chaque morceau étant supporté par un unique domaine réseau, chaque domaine étant supporté par une unique technologie de communication. Ainsi, les paramètres les plus adaptés sont utilisés sur chaque morceau de connexion et les algorithmes de spoofing les plus performants sont appliqués. Ces protocoles et architectures ont été développés avec le logiciel de modélisation OPNET et simulés avec ce même outil, ceci afin d'évaluer les avantages de notre solution. Dans cet article, nous nous intéresserons aussi à la manière de déployer de tels protocoles et architectures. La solution recommandée consiste à utiliser des réseaux actifs comme ANTS, capables de télécharger et d'exécuter du code portable dans les composants du réseau.

1. Introduction

What we call in this paper “multi-networking” is certainly a new word to qualify an operation domain not well defined and not federated by any definition or standardisation working group. This domain is looking for architectures and protocols to cope with the use of multiple networks as satellite or wireless networks having very different characteristics compared to traditional wire networks. This domain is also looking for some ways allowing data connections to serially cross several networks, or administrative domains, or Autonomous Systems (AS). This problem is especially difficult to solve, because of the arrival of some devices at network edges as firewalls or NAT servers that break the end to end IP model, and thus require mechanisms to enforce service continuity. This domain is quite important and quite a lot of work is done, but without any correlation between all actors. This paper aims to identify the main families of TCP/IP model breaking trends and the kind of “multi-networking” solutions that can be applied. It also aims to propose a way to federate them (or at least to classify them), and finally to propose an easy to deploy solution.

What we call “multi-networking” is actually related to the use of several different kinds of networks (networks being here related to OSI layers 1 and 2), and to enforce mechanisms providing the best possible quality in communications, from end to end. However, this is typically the kind of functionality of OSI layer 3 and particularly of the IP protocol. In fact, the IP protocol has exactly been designed for this purpose, i.e., to hide networks (layers 1 and 2) complexity and differences under a convergence layer in charge of all operations and decisions to manage end to end transmissions of data packets. But IP has also been designed at a time where Quality of Service (QoS) only dealt with reliable end to end connections, with no strong requirements in terms of bandwidth and delays. In addition, at that time, most networks were wire networks and even if they were different, the services they were providing were not that different. IP was then a very efficient solution to handle all these different networks and to offer users kind of a universal service, totally flat. Today, new kinds of networks are components of the global Internet. In particular wireless and satellite networks have appeared. And the most significant result of the introduction of such kinds of networks in the Internet is certainly the introduction of a wide range of new features. For example, wireless and satellite networks did introduce very different capacities, long delays in communications and high loss and error ratios. As a consequence, IP and its related transport protocol TCP that have been designed and optimised for low delays and loss ratios are not that efficient on satellite links. And these new wireless networks are providing such different services that it is almost impossible to continue using the old Internet architecture in such an environment.

This paper deals with improving the current architecture to make it capable of taking advantage of all kinds of accessible networks as ATM, mobile, ISDN, satellite, etc., each of them providing services with various QoS. On the other side, since current applications use multimedia data streams having very different requirements in term of QoS, they may take advantage of the services provided by all or part of all these networks. For instance, a Video on Demand (VoD) application can take advantage of an ISDN network for application signalling and a satellite network for audio and video. This is typically what we call “parallel multi-networking”. Our approach aims at carrying out this choice according to the QoS characteristics provided by the various accessible networks and the QoS required by applications. On the other side, connections may have to serially cross several networks as a terrestrial wire network, a satellite link and a wireless mobile network, each of these communication supports having very different capacities, delays and loss ratios. This is what we call “serial multi-networking”.

Hence, this work aims to propose a new solution relying on new protocols and architecture to cope with multi-networking. The solution for parallel multi-networking is called MMPOC-MN (MultiMedia Partial Order Connection for Multi-Networking). It takes into account application requirements and networks features. It relies on a transport protocol called MMPOC (MultiMedia Partial Order Connection), based on partial order and reliability mechanisms aiming at improving the performances of multimedia data communication. This transport protocol has been designed on the following observation: applications have to transmit their multimedia streams in respect with their required QoS. But, the famous TCP and UDP are not suited for multimedia data as audio and video (for example), because they provide either fully reliable and fully ordered, either unreliable and unordered services, whereas multimedia data requires partially reliable and ordered services. This protocol is then playing on both order and reliability to improve data transmission. In particular, it has been shown previously that such a transport protocol can help in reducing end to end delay for multimedia applications [DIA95]. MMPOC-MN also enforces synchronisation between parallel flows on separated networks. This protocol can then be tuned very precisely in order to be the optimal transmission protocol related to application requirements and network constraints.
To cope also with serial multi-networking, this protocol architecture has been extended. The new general multi-networking protocol – including both parallel and serial aspects – is called MNP (Multi-Network Protocol). It is based on the concept of splitting the end to end connection in several trunks, each trunk being supported by a single network domain, each domain being supported by a single technology. Then, the best suited parameters is used on each trunk, and the most suited spoofing algorithms is applied on data streams depending on the application requirement model.

Concerning the development of MMPOC-MN and according to the difficulties of new protocol deployment, new techniques based on the concept of active networking have been investigated, especially the one called "Mobile Code". Thus, it has been chosen to implement the protocol in Java and to use ANTS [WET98] as an easy way of protocol prototyping and deployment.

This paper is structured as follows. First, the problematic related to multimedia communications including their needs of using several networks is exposed (section 2). Section 3 then proposes a state of the art in multi-networking, especially showing how it relates to the problematic expressed in section 2. Section 4 presents our protocol and architecture for “multi-networking”: MMPOC-MN and MNP respectively. In this section, three points are addressed: (1) the MMPOC transport protocol providing a solution for multimedia communications, (2) how MMPOC-MN can take advantage of multiple parallel networks, and (3) how MNP can split an end to end connection into several connection trunks and performing spoofing operations at the edges of each trunk. Section 5 then addresses deployment mechanisms for MMPOC-MN and MNP, and for that presents the way active networks are working, and in particular ANTS. Section 6 describes the implementation of MMPOC and MMPOC-MN, presents all the used mechanisms and principles, and gives some performance measurements for these protocols. These measurements have been made using a simulation software and a MMPOC-MN prototype. Finally section 7 concludes this paper.

2. Multi-networking problematic overview

Multimedia applications require different kinds of QoS according to the types of data streams involved. This QoS is impacted by many parameters, such as reliability, throughput, end to end delay, etc. For example, let us assume that a videoconferencing application has to open 3 connections: one for transmitting audio, one for video and one for signalling. Typically, the audio stream is a CBR stream (Constant Bit Rate) with an almost perfect reliability, and a very low end to end delay to favour a high interactivity level. The video stream is a VBR stream (Variable Bit Rate) with a peak and mean rate, a minimum reliability level, a suited image definition depending on the video quality required by users (for instance a very high quality for medical imaging, and a medium or poor one for tele-teaching, etc.), and also a low end to end delay. On the other hand, the signalling connection has to be fully reliable, with a low throughput, and can work as well with an end to end delay longer than the audio or video one.
As a response to all these various QoS required, many network infrastructures using very different technologies have appeared, and can be selected by applications depending on their requirements and on the QoS provided by each network. It is now obvious that computers may have several networks interfaces. Then, they can use networks as ATM, Mobile, satellite (LEO, GEO), ISDN, etc. In the videoconference example, we can assume that the audio stream takes advantage of an ISDN connection, the video stream of a broadband VBR ATM channel and the signalling stream of a classical TCP/IP connection over the classical Internet. These networks seem to match well the videoconference QoS requirements: ISDN is a synchronous network ensuring a CBR throughput, reliability and end to end delay with almost no jitter; ATM is a broadband network providing high throughput eventually in a VBR mode, low end to end delay, and a good reliability; TCP/IP over the Internet provides a reliable service with no bandwidth reservation and no control of the end to end delay.

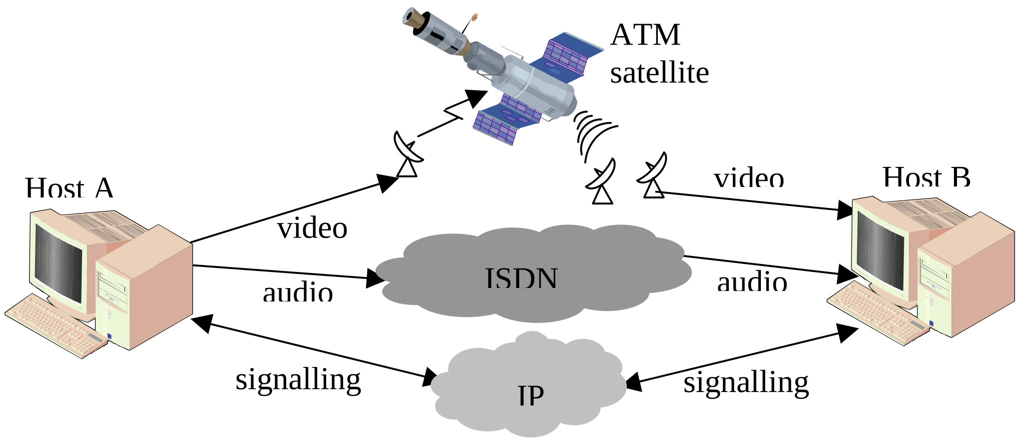


Figure 1 – Parallel multi-networking / Multi-réseau parallèle

To take advantage of all these available networks having various characteristics, a communication protocol has to be able to separate the application data streams depending on their types, and to send them on the most suited network it can access. Assuming the case of a VoD application (that is very similar to videoconferencing, except that the end to end delay constraint is looser), Figure 1 shows that the audio stream takes advantage of an ISDN link, the video of an ATM satellite link, while the signalling connection uses the Internet: this is what MMPOC-MN proposes to achieve dynamically, depending on application QoS requirements and the characteristics of accessible networks. Except the choice of the network, an important problem MMPOC-MN has to solve concerns the re-synchronisation of the 3 streams, sent on two different physical and logical networks. This is typically "parallel multi-networking".

But an other topological case can appear. In fact, it is possible that, to reach a remote user, data streams have to serially cross several networks, each having their own QoS characteristics. Depending on the type of network crossed, the implementations of the used protocol (for instance TCP) can be different or have different configurations, set specifically for each network: for example, considering the previous VoD application on a TCP service that crosses a terrestrial ATM network, an ATM GEO link and then a terrestrial MAN (Metropolitan Area Network) intranet (Figure 2), the TCP implementations and/or configurations can be different on each link. It is then very difficult to ensure the QoS through all these networks. This is typically "serial multi-networking". This kind of serial multi-networking problem is certainly one of the problems that raises the most of efforts in the Internet community. Actually, the IP protocol has been designed to unify all the communications techniques between all kinds of physical networks and their link layers. Hence, it issued the current Internet that has been managed for a long time as a flat network, the IP layer and protocol hiding network heterogeneity (for addressing, QoS, Media access, etc.) to users. And this flat management of the network has been quite efficient since the beginning of the Internet construction as network features (almost only wire networks) were quite similar or at least not as different as what we can see now. Currently, cable or fibre networks are not the only kinds of infrastructures that can be encountered in the Internet. As well, wireless and – even more difficult – satellite networks are now in the range of usable links in the Internet. Wireless and satellite networks are quite different from wire networks as they have quite different transmissions delays, capacities, and loss ratios. As a consequence, protocols and in particular transport protocols have to be redesigned, or at least tuned to maximise usage of such links. For instance, a new version of TCP has been specifically designed for satellites [ALL98]. In our VoD example, in order to maximise the usage of the satellite segment, the best solution would be to translate the transport protocol used on the wire edge networks into a protocol specifically tuned for satellite links and then providing a QoS suited to the one requested by such an application. More and more, such applicative components translating, or at least spoofing traffic are inserted in the network. Such components are said to do caching as long as they handled very asynchronous traffic, or most of the time they are now called "proxies". Proxies are then in charge of translating protocols, or spoofing traffic, but the current trend deals also with making them capable to handle streaming and / or real time traffic, that is of growing importance in the Internet [MCC00].
Another aspect that goes against the flat way of managing the Internet is related to the insertion in the network of firewalls for security. In the same way introducing NAT servers to cope with the limited amount of IPv4 addresses does not allow us to manage the connections between the two networks as a simple routing problem. In particular, NAT as well as firewalls do modify the QoS features of data streams and their temporal properties as they break the end to end IP model. Addressing what we call "serial multi-network" aims to cope with all these kinds of new problems that force us not to consider the network as flat as it was a few years ago.

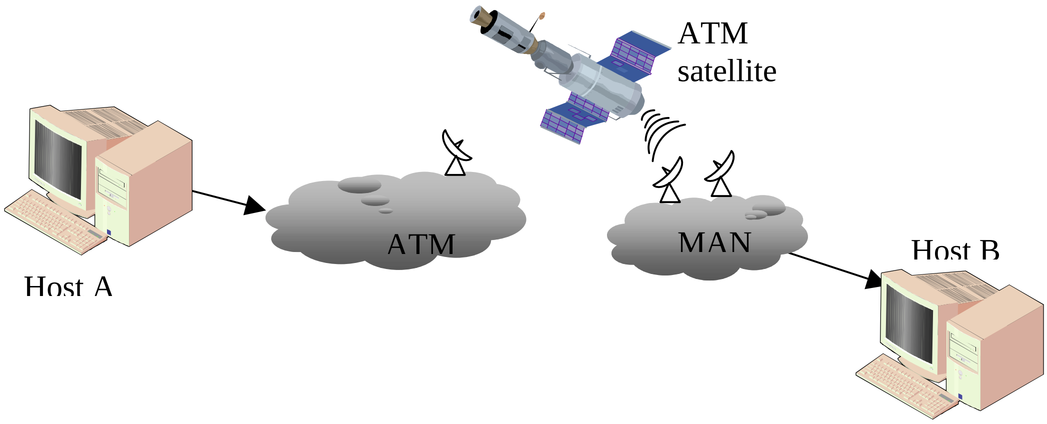


Figure 2 - Serial multi-networking / Multi-réseau série

Another example showing the need of not considering the Internet as a flat network is given by the DiffServ architecture [BLA98] that aims to provide some service differentiation, at least statistically guaranteed. This is also a specific case of serial multi-networking that raises much effort in the Internet community. The starting hypothesis in DiffServ consists in considering that QoS is easy to achieve on a single domain that is uniformly managed (as an ISP or carrier network). The point then deals with enforcing QoS guarantees between all domains that are crossed by packets and flows. At the opposite of caches and proxies, this serial multi-networking is handled at the TCP/IP level instead of the applicative level.
Of course, the VoD application can also have to cross serial and parallel composition of different networks as on Figure 3. On this Figure, audio stream takes advantage of an ISDN link, the video stream crosses a terrestrial ATM link, a geosynchronous satellite channel and then a metropolitan intranet, while the signalling stream takes advantage of the classical Internet.

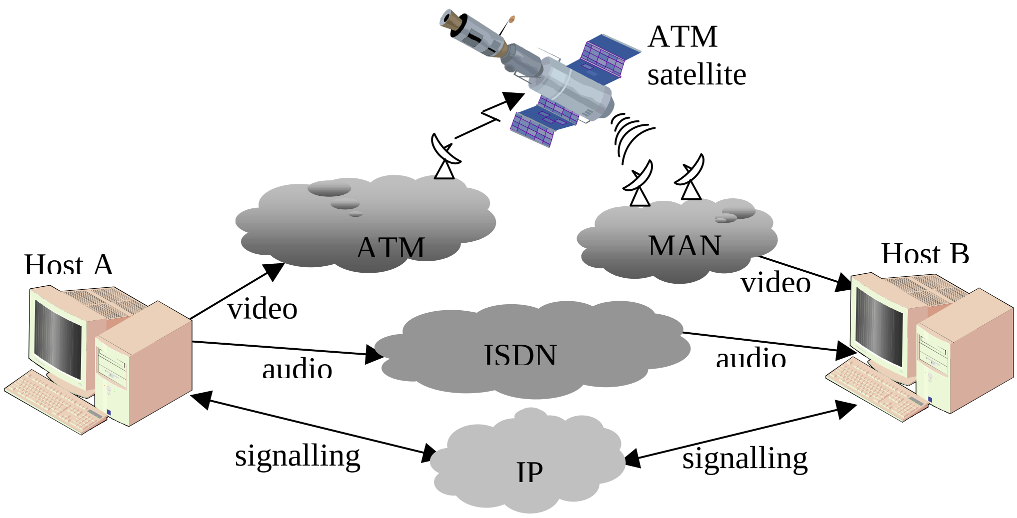


Figure 3 - Parallel & Serial Multi-networking / Multi-réseau série et parallèle

3. State of the art in multi-networking

Previous parts showed that many situations are more or less closely related to multi-networking issues. And this status is even more true in the case of serial multi-networking. This section then aims to describe some work that is currently in progress. In this part, parallel aspects are discussed first, followed by serial ones just after.

3.1. Parallel multi-networking

Today, only a very few architectures allow the use of several different networks in an integrated way. Nevertheless, this concept can appear, for example, in some distance learning environments that use analogical communications for audio and video, and digital ones for data exchange and control sessions. But these environments are often limited to short distance. Some work are also performed in the domain of mobile communications; their aim consists in selecting a new network if the one the application was using is temporarily not available (because of a handoff problem for instance) [STE98]. This work also concerns environments allowing the selection of the network interface users want to use to send data [INO97][LAN98]. But generally these systems do not take into account the complete set of QoS parameters required by applications, and only focuses on few ones as physical criteria (quality of received signal, cost, etc.), and most of all are limited to the use of a single network interface at a given time.

3.2. Serial multi-networking

Finally, it appears that very few work dealing with parallel multi-networking has been done at this time, maybe because most of users do not have multiple network access yet. On the other side, much work has been done to deal with serial multi-networking, even if there was almost no coordination between all research and engineering activities. Thus, many specific solutions have appeared in the domain of caching and proxies. Since few time, an IETF working group (called PILC: Performance Implication of Link Characteristics) started working in that area. This group then wrote a proposal draft where they recommend some methods for splitting end to end connections and some mechanisms for proxies, going in the same way as our previous work on MMPOC-MN and MNP. Next part will be describing this working group proposal. Another example of a solution that deals with serial multi-networking by changing the Internet architecture to avoid changing networks is provided by caching mechanisms as well as the CDN (Content Delivery Network) approach for the Internet. Their solution will be described in 3.2.2.
Nevertheless, the multi-networking problem related to the introduction of firewalls and NAT servers in the network is not addressed at all. However, with such kinds of devices, the multi-networking issue is much more difficult to handle as the end to end IP model is broken. The second part of this section is also detailing the firewalls and NAT problematic. Of course, our MMPOC-MN / MNP solution described in section 4 is able to cope with such devices, as it will be demonstrated later.

3.2.1. ITEF WG "pilc" proposition: Performance Enhancing Proxies (PEP) and other tracks recommendations.

Performance Enhancing Proxies (PEPs) are often used to improve degraded TCP performance caused by some characteristics of specific link environments, for example, in satellite, wireless WAN, and wireless LAN environments. Different types of PEPs have been described as well as the mechanisms used to improve performance in [BOR01] with a special study related to proxies operating with TCP. In addition, motivations for their development and deployment are described along with some of the consequences of using them, especially in the context of the Internet.
Note that this draft does not make any recommendation, for or against, with respect to using PEPs. Standard track recommendations are developed for individual link characteristics, e.g.,

links with high error rates, links with low bandwidth, links with asymmetric bandwidth, etc. by the Performance Implications of Link Characteristics WG (PILC) [PILC01].
The main mechanism studied in [BOR01] to improve communication performance is related to TCP ACK handling. These ACK handling mechanisms which may be used separately or together are:

- TCP ACK Spacing (described in [BPK97]) to avoid TCP bursts of data segments starting when the sender receives a burst of TCP ACKs;
- Local TCP Acknowledgements sent back locally by PEP are very useful when the underlying network has a high product bandwidth $\times$ delay;
- Local TCP Retransmissions done by PEP when the data is lost on the following path between PEP and the receiving end system;
- TCP ACK Filtering and Reconstruction used to avoid congestion problems that may appear when the back stream of TCP ACKs grows up on a highly asymmetric link so that congestion may arise.

Local TCP ACKs and retransmissions must be used when splitting end to end TCP connections. On a split connection, the most negative effect is the break of the end to end semantics, which is one of the main architectural Internet principle [SRC84, CAR96]. Nevertheless, this approach is very useful for us, especially if the Transport protocol used in each hop, is not restricted to TCP (because of its small capabilities in term of QoS).

3.2.2. Caching and address translation

As it has been said in section 2, several well known functions in the Internet deal with serial multi-networking issues. This is the case for caching mechanisms, that have been contributing to the Internet performances improvements. Caching is a two-folds activity: first network capacities are used to download large number of web pages during night hours (when bandwidth availability is maximized) starting with the ones that received the greatest number of hits. The choice of pages to download is then most of the time based on long range statistics. On the other hand, and to cope with the high rate of information published on the web, caching is also used to record web pages that are interactively downloaded on user request. The method chosen by caching to cope with multi-networking issues is to locate the information as close as possible from users. Cache servers are located on the same network as users, i.e., on the ISP network. This caching mechanism fulfils its requirements as long as web pages do not change several times per day. Otherwise, it can happen frequently that the pages stored in the caches are not up to date. Nevertheless, the idea under the caching approach consists in changing the initial web architecture. It aims first to reduce the traffic outside the considered network, and then avoids to cross peering points that are generally part of the bottlenecks in the global network. In fact, this approach is very “handsome” as it solves the multi-networking problem by cancelling all multi-networks connections. Of course, this solution only works for asynchronous traffic. It cannot match real time traffic requirements.
CDNs as Akamai have a similar goal, except that it also moves the web servers from edge to core network to avoid the bottleneck related to the limited capacity of access networks and peering links.
However, it is clear that even if such techniques have a strong impact on traffic reduction on the global Internet, they are changing the typical architecture of the Internet. In particular, one can wonder what is the limit of the CDN’s system? CDN have an approach with many similarities with VPN. And the VPN feature is very important for performances. If too many servers are located in one VPN, and if too many users attempt to access it, the VPN would lose its QoS properties, and all the system would fail.

Some other components in the Internet as NAT servers as well as firewalls deal with multi-networking issues. NAT is in charge of transmitting data packets from one network to another that does not work with the same addressing policy. In the same way, firewalls are in charge of enforcing secure access from a public to a private network having different security policies. They are both good examples of the wide range of functionalities in the Internet that deal with multi-networking. They are also good examples that show that the QoS of data streams can be disturbed by such components breaking the end to end IP model. For instance, it is well known that it is quite impossible to start a high quality videoconferencing session between users that are located behind a firewall. This is the kind of issues this paper aims to address.

4. MNP: a Multi-Network Protocol

Given these new needs and constraints in terms of multiple networks available and given the very limited contribution done until now in that area, we have been working on the extension of a multimedia transport protocol towards a communication protocol suited for taking advantage of multiple networks.
In this section, our protocol is first defined, and then the two main cases of multi-networking solutions are described. The benefits of these two cases are illustrated in section 6 thanks to simulation results obtained with the OPNET modelling Tool.

4.1. General presentation

According to the multi-networking problematic, described in part 2, the communication protocol has to be able to adapt itself to the various kinds of QoS provided by the underlying networks. Our interest does not only focus on end to end communications, but also on features of every hop. Each hop has to take maximal benefit from the service provided by the underlying network, where a hop deals with going from an IP domain to another. This is very challenging: what kind of protocol may be able to fit to wire optical fibre network as well as a wireless link or a satellite channel in order to guarantee users end to end QoS? Closely involved in this question, two problems must be solved:

- the multi-network protocol must be able to choose the best available network, mapping the required application QoS on the provided network QoS;
- the multi-network protocol must be able to guarantee end to end behaviour, using several consecutive split connections, each hop using a different instance of network QoS; here, an additional mechanism must be available to control transfer from the previous serial connection to the next one.

All these problems must be considered in our Multi-Network protocol (MNP). This protocol can be seen as a basic communication protocol for multimedia data with the enhancements needed to address parallel and serial multi-networking communication. As described in [BOR01], in terms of layering, these specific mechanisms added to the protocol may take place either at the Transport or at the Application Layer. Our protocol is even more difficult to locate in classical network architecture, layering both on Transport and Application. The end to end semantic of the classical TCP/IP layer is broken and a new functionality is needed at the edges of network domains (network domains having here the same definition as in DiffServ) to be able to send data packets from one domain to the next until the final receiver. The breaking point is implemented using a special device, including a classical IP router functionality enhanced by Transport and Middleware entities as shown in Figure 5.

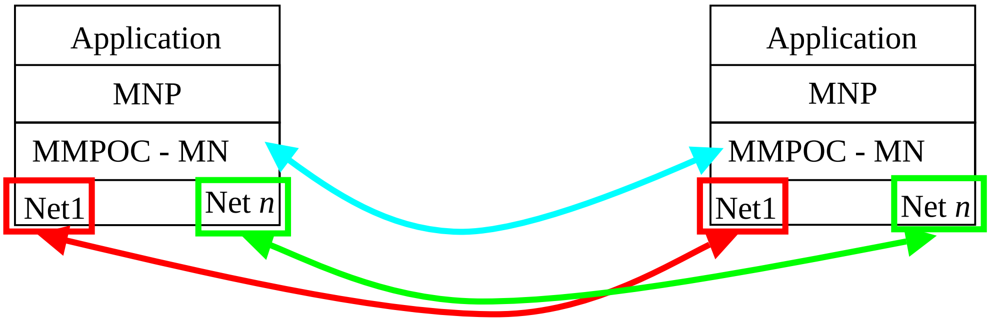


Figure 4 – MNP architecture for parallel multi-networking / L'architecture MNP pour le multi-réseau parallèle

Along a parallel MNP connection, the connection is set up from one end to the other. The protocol must be able to choose among the available accesses the one that better fits the application requirements (Figure 4). Applied to multimedia communication, using separate connections relying on various kinds of networks requires to resynchronise the monomedia streams before delivery to upper layers. This synchronisation may be logical only, and not temporal, since applications are more suited for final temporal synchronisation [OWE98].

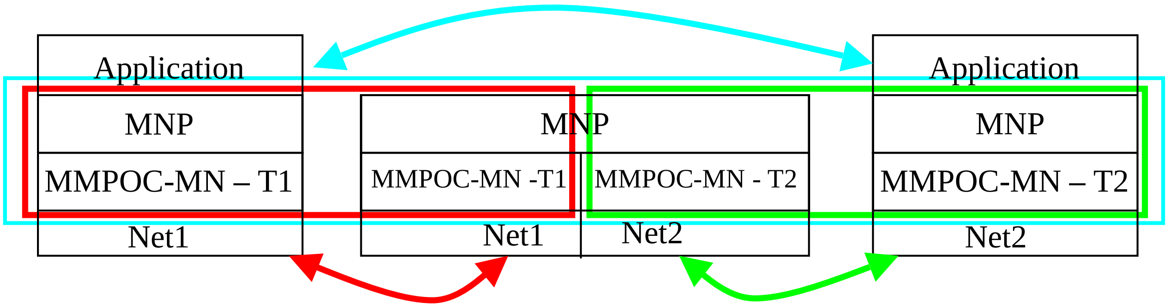


Figure 5 – MNP architecture for serial multi-networking / L'architecture MNP pour le multi-réseau série

With MNP, the end to end connection is achieved by serial hop by hop connection; on every hop, the MMPOC-MN protocol is used. It is a multimedia transport protocol based on partial order and reliability described in the next subsection. The end to end argument is preserved by MNP functionalities, including end to end QoS guaranty (as shown Figure 5). These functionalities will be defined further in this paper.

4.2. MMPOC: A multimedia transport protocol based on partial order and reliability

It is now well known that the famous TCP and UDP Internet transport protocols are not suited for multimedia data streams of distributed applications, because they cannot support the various required multimedia QoS. In fact, TCP only provides a fully ordered and reliable service, UDP an unordered and unreliable service, while multimedia data streams require a partially reliable (in fact, it is not troublesome if an image of a 25 images/s video stream is lost or damaged), and a partially ordered service (because multimedia applications have to handle several serial and/or parallel compositions of data streams). As a solution, a partial order transport [AME94][DIA95] is a transport that delivers data units sent on one or several connections, following a given (partial) order. This order is any order between the total order

(as TCP) or no order (as UDP). It can be expressed as a serial/parallel composition of sub-objects or data units. Note that this order can be for instance described by a Time Stream Petri Net – TSPN (a temporal extension of the general Petri Net model) [OWE98] – as on Figure 6 that represents the serial/parallel composition of sounds and images of a videoconferencing application. In this case, this delivery order can be seen as the logical synchronisation of multimedia objects, synchronisation being one of the essential key-points of multimedia distributed systems.
Furthermore, the possibility of losses in networks leads to the interesting notion of partial reliability. This notion is tightly related to the transport QoS : it defines a nominal QoS and a minimal QoS under which the user requested service is no more ensured. This minimal QoS can be expressed by defining a set of acceptable losses, for instance a maximum number of losses inside a sequence, a maximum number of consecutive losses,… When an acceptable loss can be detected (i.e. when a received object has a numbering higher than the expected one), the missing object can instantaneously be declared as lost (i.e. leading to earliest indications of losses), and the (next) data already received can be delivered at once to the application (i.e. leading to earliest deliveries). If the loss cannot be accepted in terms of requested reliability, retransmission will occur.
The required partial reliability, defined in the partial order transport service and resulting in earliest loss indications and deliveries, is deduced from the application requirements [OWE98].
In fact, two approaches exist for managing partial reliability: media per media and by groups of media. Media per media means that the receiving stream entity can only handle the partial reliability on the stream it manages, and not on the other streams of the multimedia connection. In the per group of media management, the receiving entity can declare losses on other streams of the same multimedia connection, which leads to a more interactive behaviour, as we will see now.

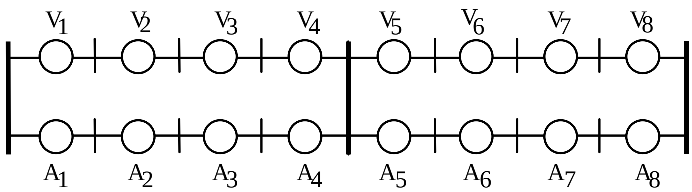


Figure 6 - Partial order example / Exemple d'ordre partiel

Let us consider Figure 6 that represents a Petri Net of a serial / parallel composition of a multimedia connection (audio and video of a videoconference application for example). Let us assume that the maximum allowed number of losses on each stream of a synchronisation period (V1-V4, A1-A4) is of one. Let us also assume that data V1, V2, A1, A2, and A3 have been received by the receiving entity and delivered to the application.
Let us then assume that the remote entity receives V4 ; the strong fulfilment of the partial order on the video connection says that V4 cannot be delivered to the application : this is because it is logically located after V3 and V3 has not been received yet. Now, V4 can be delivered if V3 has been lost, because of the selected reliability (one data allowed by period) ; this is the principle of partial reliability. As one loss is acceptable on each period for each stream, and in order to deliver V4 as soon as possible, V3 will be declared as lost, and V4 is delivered. This is the principle of earliest losses and deliveries.
Let us now assume that V5 is received. Regarding the partial order, this object cannot be delivered, because it is logically after A4 (because of the inter-streams synchronisation

following A4 and V4). With a media per media management of the partial order, V5 has to be stored until A4 is received or is declared lost.

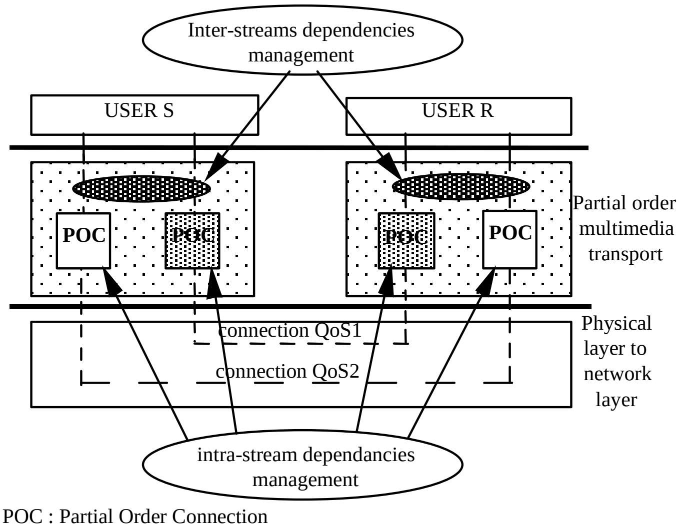


Figure 7 - Architecture of a partial order transport / Arcitecture d'un transport à ordre partiel

However, with the per group of media solution, the manager of the video connection becomes able to create losses on the audio connection. In this case, as one loss is acceptable on the audio stream (during a period), it will declare A4 as lost and delivers V5 to the application almost as soon as it receives it. This by group of media partial reliability management leads to the design of the highly interactive architecture given in Figure 7, which also shows the multi-connections management [OWE98].

So, if multiple accesses to various networks providing different QoS are available, using MMPOC allows us to easily set up monomedia connections on the most convenient network with regard to the considered media requirements. The synchronisation mechanism provided by MMPOC still works if several network accesses are used. This enhanced version of MMPOC is MMPOC-MN, described in the next section 4.2.

On the opposite side, if the end to end connection is broken in consecutive hops, benefits can be gained from using this protocol for each hop, the protocol being tuned to maximise its performance on each hop. QoS may be defined hop by hop, mapping locally MNP middleware constraints on local underlying network QoS. MNP is described in section 4.4.

These two main features show that MMPOC provides a very interesting functionality, needed in MNP for both serial and parallel multi-networking. This will be detailed in the following subsections. The complete multi-networking solution involves both serial and parallel configurations. Then, the hop becomes more complex and may be addressed using parallel multi-networking, split connection may be put together to maintain end to end semantics thanks to MNP functionalities and that have to be deployed at network edges.

4.3. Parallel multi-networking: MMPOC-MN architecture

In order to provide a solution to dynamically select the most suited network related to the QoS required by each stream separately, MMPOC-MN, extension of the MMPOC protocol, is proposed (Figure 8). This extension allows the selection of the most suited network for the transmission of each data stream depending on the QoS characteristics of application data streams and on the knowledge of the accessible networks (from the sender but also from the receiver).

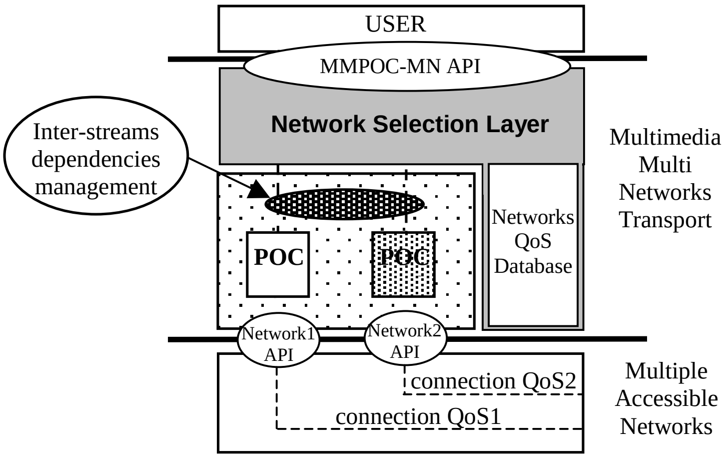


Figure 8 - MMPOC-MN architecture / L'architecture MMPOC-MN

The main problematic of such an architecture consists in selecting the right network. For that it is first needed to determine the QoS parameters of each accessible network. Then a study has been conducted on communication networks and in particular on new ones as GEO/LEO satellites, Mobiles, etc. It appears that a set of generic parameters can be found out for the characterisation of the QoS provided by these networks. They are given in Table 1.

	Throughput	*RTT*	*Gigue*	*BER*	*Security*	*Broadcasting*	*Cost*
ATM/CBR	Guaranteed	Small	Small	Small	Yes	No	High
ATM/UBR	?	Small	Small	Small	Yes	No	High
ISDN	64 Kb	Small	Small	Small	Yes	No	Low
Satellite	> 64 kb	High	Small	High	No	Yes	Middle
Mobile	< 64 Kb	Small	Small	Very-High	No	No	Middle
Internet	?	?	High	High	No	No	Low

Table 1 - QoS provided by some network services / QoS fournie par quelques services réseaux

It clearly appears that some parameters are not significant for some networks: for example, guranteeing bandwidth for a best effort network as an IP network. But this information remains interesting and can help MMPOC-MN in its network selection: thus, an IP network is not suited for a data stream (video or audio) requiring a guaranteed throughput.

All these parameters allow the building of a database that contains all the characteristics of available and accessible networks. It is however to note that at the opposite of what is described in table 1, the network database indicates precisely the value of the different parameters, and not only an indication. These QoS characteristics belong to two distinct categories: the first concerns the characteristics provided by the system that support the network drivers, while the second consists of the parameters that have to be measured (as the end to end delay for example) in an experiment phase. The network selection can then be done depending on general network characteristics, but also depending on its state.

Then, to select the most suited network for one data stream, MMPOC-MN uses a pre-defined metrics that measures the distance between the requested QoS for the data stream and the one provided by each accessible network, and then selects the one with the lowest distance.

4.4. Serial multi-networking

4.4.1. Current use of various network providing different QoS

If the Internet is currently deployed on the top of various kinds of links, it is always transparent for the upper layer. End to end communication is usually involving TCP or UDP, whatever kind of underlying network is used. But this approach is hardly restrictive. For instance, TCP relies on RTTs to set up sending window size. If a satellite link is involved in the connection, it introduces a high product "bandwidth × delay". For one hop only, from earth to satellite and then back to earth, the additional delay is around 250 ms. So it has been shown that TCP is unusable in such a configuration [PAR97]. Nevertheless, the main advantage of using TCP for distributed application programming is that the programmer can work without any knowledge of underlying networks. To improve the performances of this facility, the idea is to split the end to end connection in several hop-by-hop connections. The goal is to map the used hop-by-hop transport protocol to the network characteristics. This proposal has also been included in PEP [BOR01], but only for TCP users.

4.4.2. MNP architecture

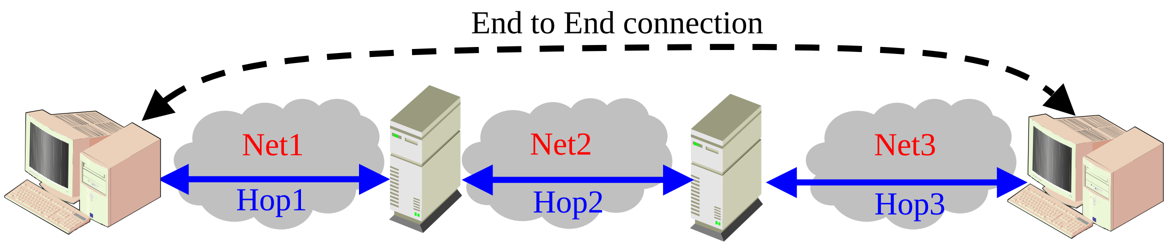


Figure 9 – Serial multi-networking configuration / Configuration pour le multi-réseau série

Let us consider the configuration given on Figure 9. And let us remember that, as said before in section 3.2.3 devoted to PEP description, the main mechanism is related to TCP ACK handling with:

- TCP ACK Spacing;
- Local TCP Acknowledgements;
- Local TCP Retransmissions;
- TCP ACK Filtering and Reconstruction.

Using end to end TCP is not interesting for us, as shown by the previous description of multi-networking. But local TCP ACKs and retransmissions must be used to improve communication performance. This advantage reduces the drawbacks of splitting end to end TCP connections, inducing a break of the end to end semantics. MNP is not necessarily using TCP (it is using MMPOC-MN, but TCP is a specific case of MMPOC-MN where there is only one parallel connection, with total order and reliability, and also using a single network support). So, to design our Multi-network serial protocol, there is only to address the main problems induced by:

- breaking end to end semantics (MNP problematic)
- trying to improve protocol performance (MMPOC-MN problematic).

Let us consider first the consequence of breaking end to end semantics. What will happen in the configuration given in Figure 9 if the message is lost on the last connection (3)? This message has already been acknowledged and the initial sender has thrown away the message from its sending buffer. If this retransmission has to be transparent to the initial sender then

the last gateway crossed by the message must have kept a copy of it. Here it is obvious that a proxy/cache mechanism is needed, to guarantee an end to end behaviour of the split connection.
Now, if a slow link is encountered along the split connection, then the end to end ACK is delayed, even if the other links are very fast. The retransmission has to be done from the source, re-crossing the slow link; and the total delay is heavily increased each time the slow link is crossed.
The solution to improve the global performance of such a connection is to split this connection, setting one hop per crossed network. Benefit is gained from local mechanisms implementation (acknowledgement and retransmission) if the gateway (the equipment between two consecutive hops) provides retransmission buffers. Then, the retransmission is done only from the previous equipment (gateway or end user computer), reducing the delay to re-send the requested data.
Splitting the connection needs to solve the two following problems:

- deployment: the implementation of this architecture depends on deployment facility provided by the different involved networks;
- end to end QoS: the definition of each hop QoS is not sufficient to guarantee the end to end QoS requested by the application; coherence is maintained from one hop to the next by an additional part of the designed architecture to be set into gateways.

These questions related to protocol deployment are not obvious, but it will be shown in the next section how this deployment can be done. Some new concepts such as active networks will be helpful. The architecture proposal does not concern the core equipments (no code to be deploy there) but only the edge gateways (different proposals of such equipment have been reported [6WI01]). All this will make possible the deployment of MNP in a very close future.

5. Deployment

Respecting our multi-networking classification and our related solutions, this section proposes to examine how they can be easily deployed over the Internet. Then, in this section, the first part deals with the parallel multi-networking paradigm, and it is shown why this problem can be seen as a software distribution problem. Finally, a more generic and flexible protocol architecture is presented, which provide an integrated solution for the whole multi-networking problematic.

5.1. Parallel case: Protocol deployment

Implementing new transport protocols is really interesting if it can be widely and easily deployed. In fact, following the work achieved on the implementation of MMPOC using the Sun Solaris 2 streams mechanism in the operating system kernel [FOU96], it appears that in spite of the performance gains induced by such a protocol implementation, its deployment over an heterogeneous platform is very difficult. This lead progressively to give up this way of development which deployment and maintenance were too difficult and costly. Thus, even if this protocol brings interesting performances improvements, its success is tightly linked to its ability to be easily ported and deployed world wide… what is definitely not the case.
An actual solution, is to get the specific protocol stack when subscribing to a new service, as for instance satellite Internet, or WAP. If an host intends to communicate to an other host within the same network, there are many chances that they are owning the same communication stack version, and so can use it to communicate. But, when contacting a host in an other network, nothing guarantees that they could find a common version of their

communication stack. A WAP browser can only access a WAP Web server, which implements the same WAP protocol stack.

This statement has lead us to in depth investigate new solutions and new languages, providing code mobility capabilities over heterogeneous networks.
Environments providing code mobility mechanisms seemed to be well suited to the problem of transport protocols distribution and deployment. Much work was running in this domain, but very few platforms were available [JAV98, JAV99]. However it is important to note that the platforms based on the Java programming language were the most advanced in terms of performance and portability. Then, the solution proposed by the HotJava environment appeared as the most suited for our MMPOC-MN distribution [BER00].
HotJava is the web browser developed by Sun to demonstrate the Java capabilities, and the new abilities provided by the programming of applications in Java. Its ability to be extended dynamically, and to integrate automatically new behaviours, is a strength that differentiates HotJava from other browsers. HotJava is capable to automatically download Java code, visualisation programs and new protocols that can be integrated in the browser, and this in a transparent manner for users. In HotJava, the protocols and viewers are implemented in Java, as well as all their extensions.
A HotJava document is then referenced by an URL and its associated protocol allowing users to access the files on the remote workstation. If the protocol is not implemented on the machine, the code distribution mechanism provided by HotJava allows the downloading of the protocol byte code from the server. This mechanism thus provides a solution for deploying new protocols as protocols for multi-networking.
This environment then has allowed the deployment of the MMPOC-MN protocol, as well as a VoD demonstration application.

5.2. Serial case: Protocols and proxies deployment using active networks

To deploy the MMPOC-MN / MNP solution proposed in this paper, we are looking for a new architecture that integrates in the same way the parallel and serial multi-networking solutions deployment. In fact, the HotJava solution is not powerful enough and is not able to handle proxies deployment. As a consequence, a new deployment methodology has to be found. Active networking seems to be a promising solution to dynamically deploy some programming code, and is especially well suited for protocol and proxies deployment.

The « Active Network » concept is a new way of thinking and may be one of the solution for the lack of flexibility and expandability of classical networks. The main goal deals with allowing dynamic integration of services, and thus to be able to adapt networks to their future utilisation, instead of anticipating which is evidently impossible. Three complementary responses to this problem appeared [TEN97]. The programmable switch approach (1) [ALE98] maintains the existing packet/cell format, and provides a discrete mechanism that supports the downloading of programs. Separating the injection of programs from the processing of messages may be particularly attractive when the selection of programs is made by network administrators, rather than individual users. In contrast, the *capsule* approach (2) [WET98, SCH99] goes somewhat further – the passive packets of present day architectures are replaced by active miniature programs that are encapsulated in transmission frames with data and executed at each node along their path. Finally the *mobile code* approach (3) [HAR96, JAV98, JAV99] gives an easy way to compute programs in an heterogeneous environment because it allows programs mobility across a network.

5.2.1. ANTS presentation

ANTS (1999, D. J. Wetherall, MIT) is a Java implementation of the capsule approach, under the form of a set of utilisable Java class with a well defined API. The basic idea is the following one : not only data are sent through the network, but data with the associated processing code. In classic routers, all packets are treated identically whereas, in active nodes (routers in an active network), each packet contains its own handling instructions.

An ANTS network is composed of these main entities :

- *Capsules* are special types of packets that encapsulate data and the processing code for these data
- *Nodes* replaces classic routers, they execute the code embedded in each capsule
- *Channels* are used to link nodes together
- *Applications* use nodes to send capsules through the network.

An ANTS network service is represented by a set of user-defined capsules. This service is also called an ANTS *Protocol*. New network services can be expressed in terms of capsules, and must be deployed within the network infrastructure before they can be used. ANTS provides an automatic code distribution system that makes it straightforward for an application to begin using a new service. The ANTS dynamic code deployment system transfers the code that implements a new service to nodes along the path that a capsule (using the new service) follows (Figure 10). The code automatically transferred is cached for later use.

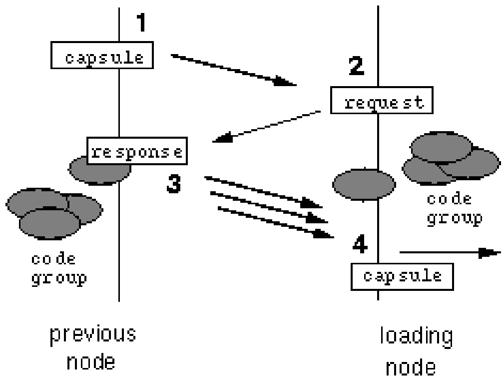


Figure 10 – ANTS code deployment / Déploiement de code avec ANTS

5.2.2. Protocol deployment

The new proposed architecture integrates a way to dynamically deploy the MMPOC-MN / MNP protocols through and inside the network. A capsule active network (as ANTS) is a fine solution to provide a way to dynamically deploy protocols and serial multi-network gateways. Protocol code deployment is provided by the Java based implementation, respecting the HotJava principles. Protocols and code groups (in the ANTS meaning) provide a way to add user defined processing in the network, and as a consequence to dynamically deploy serial multi-networking gateways.

The MMPOC-MN / MNP protocols implementations have been performed in Java and on top of the UDP/IP service. This choice has been made first for the ability of this language to produce portable code, in order to ease the deployment using active networks. First of all, a

Java performance evaluation has been achieved to validate this choice. Thus, the Java socket API performances as well as the threads switching mechanisms have been deeply studied. These studies and evaluations have shown that the performances strongly depend on virtual machines, but globally the performance speedup between a program testing these aspects written in C and the same written in Java is lower than 5. These results have after that been confirmed by [KRU98] and then they show the ability for Java to be used for communication protocols implementation Given these performances and the growing power of workstations and PCs, implementing protocols in the user space of the operating system is no more an heresy.
Then, the protocol is an ANTS application, using network probing capsules to choose the right network according to application QoS requirements. Two capsules types are defined, one to measure the end to end delay over each networks, and an other one to evaluate the instantaneous available bandwidth with the packet pair method [SRI91]. The database that contains all the characteristics of available and accessible networks is then complete.
An ANTS application is not transparently deployable as an HotJava protocol, but thanks to the Java portability the code maintenance is easy and the protocol can be downloaded and rapidly installed.

In order to deploy our protocol, let us assume that the considered networks involve specific interconnection devices. These devices may be active ANTS nodes working on a Linux PC closely to a classical IP router or a dedicated new equipment such as 6Wind edge device [6WI01]. These devices are able to work as classical IP router, but also as a programmable node, allowing remote code loading and running.
Now, two deployment methods have been investigated:

- Static deployment. The protocol and proxy codes are loaded by network administrator. They remain available and are only invocated on request when the connection is set up.
- Automatic deployment. When the connection is set up from on end to the other, ANTS capsules get through the whole path, and the protocol and proxy code are loaded in each crossed node. If the node needs to work as a proxy, the protocol set-up will configure the proxy and the code will remain available. Otherwise, the proxy code will be unloaded.

Nevertheless, as it has been said before for serial multi-networking, nodes locations where the spoofing is done, have to be chosen with a great attention to obtain a performance gain: generally before a wireless link, i.e. at a wireless network edge router, or before a slow NAT server. For that, an *active node* has to be configured to allow the spoofing code to be downloaded and run at this location. Currently, only programmable routers can host an active router that can be, for instance, the Java code of the ANTS platform. Our global architecture requires to construct a virtual network of ANTS active nodes on the top of an IP network. Then, when a connection opens, if one or several active nodes on the path are crossed, spoofing code can be downloaded and activated on these active nodes.
This deployment work is the main part of the work remaining to do.

6. Measurements

To confirm theoretical concepts and architectural design of the MMPOC-MN protocol, some measurements have been done. To evaluate the performances of such communication architectures, simulations have been performed using the OPNET modeller. OPNET is a network development environment for the design and analysis of networks, network equipment, and communications protocols. In order to confirm the results obtained using the simulator, measurements have also been done with an implementation of the MMPOC-MN protocol and a VoD application.

6.1 Parallel Multi-Networking Results

The parallel aspect of the multi-network protocol brings two sorts of improvements that should be studied separately for better understanding.

6.1.1. Mono-media results

First, the selection of the network that better fulfils the application requirements from accessible interfaces provides a qualitative change. It is obvious that a synchronous network for carrying voice is most suited than a best effort network, or that choosing a broadband satellite network instead of a phone link to download a movie file is a good idea. So, it is not significant to do such comparisons, and to quantify them… At least for our concerns.
On the other side, the gains induced by partial reliability that avoids unnecessary data retransmissions (according to the application specification) can be easily measured. Figure 11 presents simulation results for two streaming applications over geosynchronous satellite. The experiment configuration is the following: an end to end satellite link is used, the average Bit Error Rate is 10^{-7} with a 500 ms round trip time. Two types of traffic are studied: a 25 frames/s MPEG video stream, and a H263 Hi-fi audio stream. The QoS application requirements are for audio, a fully reliable connection, an for video, a 90% partially reliable connection. The end to end delay is then measured.
The quantity of stream samples (in percentage) received before a given delay (abscise data) is presented on Figure 11. This figure shows the cumulative distribution function of the delays experienced by each frame between the sender and the receiver. For instance using a 90 % reliable service, it has been observed on the MPEG streams that 85 % of packets experienced an end to end delay lower than 1 second. As a comparison, with 100 % reliability (as TCP for instance) only 65 % of MPEG frames have a delay under 1 second.
Concerning H263 frames, delays are much lower. Actually, H263 induces smaller packets (around 100 bytes), so the probability to have a bit error on this packet is reduced compare to large packets. The IP layer then notices less corrupted packets to be discarded in a H263 streams than in a MPEG stream (whose packets are around 1500 bytes).
This information on loss rate depending on the kind of applications and formats used is very useful to integrate in the MMPOC-MN database and is of great help in the network selection process when several networks are available.

We also showed that the partial order transport protocol helps in reducing the average usage level of receiving buffers (Figure 15).

In addition, the MMPOC-MN synchronisation mechanism allows the protocol to reduce even more the end to end delay, in reducing waiting time during streams synchronisation phases, streams being sent on two separated network infrastructures.

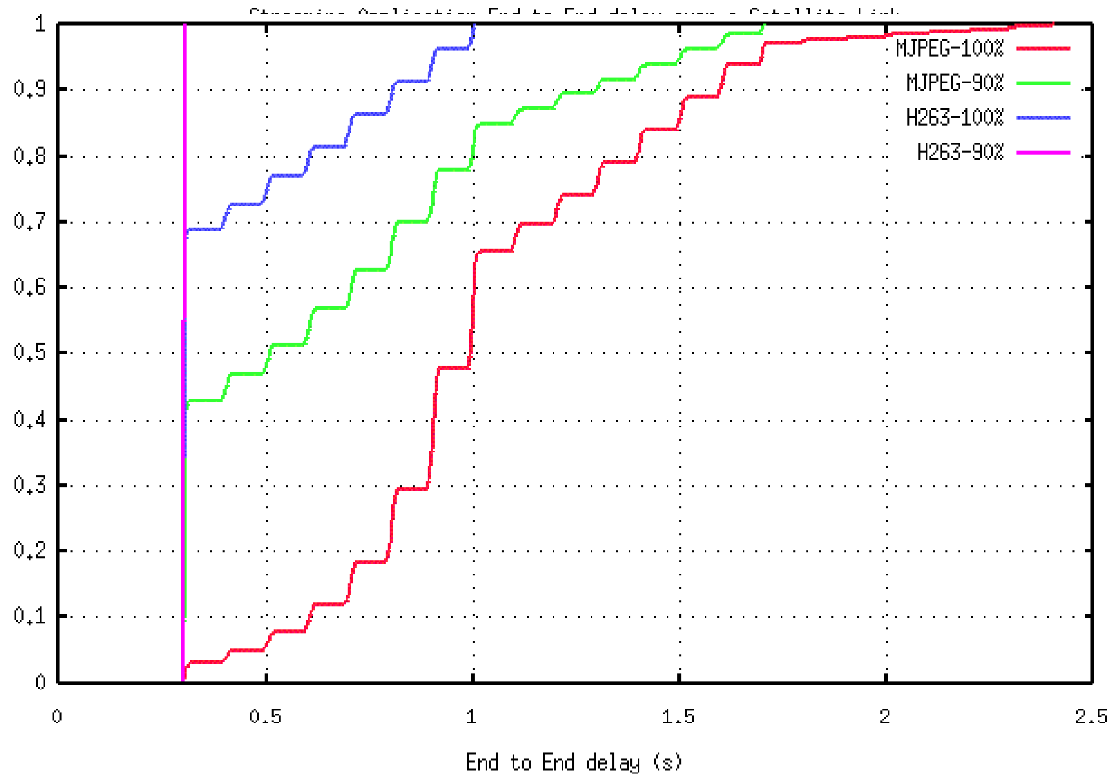
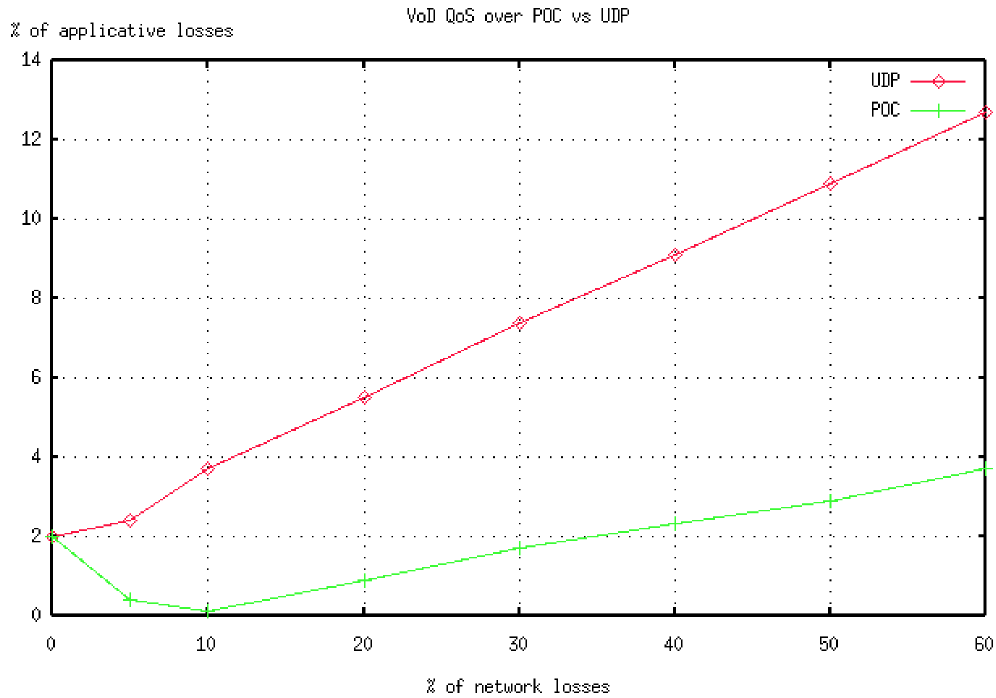


Figure 11 – Partial reliability for streaming over satellite links / Fiabilité partielle pour des flux sur des liens satellites

6.1.2. Applicative reliability improvement

After having demonstrated that MMPOC helps to reduce end to end delay, this part shows that any application using MMPOC-MN can get a better reliability than using UDP. In fact, we want to show that the service provided by the MMPOC-MN service is better than the one of the basic network.

It is important to mention here that the VoD application that has been designed and that is used in this experiment, integrates audio/video synchronisation mechanisms as described in part 4.2 and [OWE98]. Also note that the use of a partial order transport (thanks to its earliest delivery and loss mechanisms) avoids to waste time, waiting for lost or late data, as far as the partial reliability is not violated. Such time wastes are really hurtful for QoS. In fact, because of the synchronisation constraints to respect, the application has to discard some objects (images or sounds) when they are too late and then when the application has not enough time to compute them before their temporal deadline. Thus, in this section, the losses at the application level will be measured in two cases: (1) when a partial order transport and (2) when a classical connectionless transport service (UDP) are used. Of course, one can ask why we chose to compare POC with UDP instead of RTP that is the transport protocol recommended by the IETF for real-time multimedia applications? The reason is simple: RTP relies on UDP. If UDP can be considered as an empty protocol – as far as QoS issues are considered – RTP does not provide a better QoS than UDP. Even worse, RTP handles a timestamp for every packet, and if the receiving RTP entity detects that a packet is too much



delayed, then this packet is automatically discarded (not delivered to the application). Thus, we chose to make the comparison with UDP, as UDP provides a better QoS (in terms of bandwidth, loss ratio, and delays) than RTP. So, if POC provides a better service than UDP, it is consequently better than RTP. To perform this comparative evaluation between POC and UDP, a network simulator allowing to simulate losses on the network has been used. This simulator is a Solaris 2 stream module, located on the sending machine (between the sending process and the ATM driver) that discards some packets. Results of measurements are described on Figure 12. It shows the applicative loss average in the 2 cases (using UDP and POC) depending on the network loss level. In this case, no retransmissions are requested, and then Figure 12 only shows the applicative losses due to synchronisation mechanisms, that have to be added to the network losses induced by the network simulator. The support network is a 155 Mbps ATM network.

Figure 12 - VoD QoS over POC vs UDP / Comparaison de la QoS pour la VoD sur POC et UDP

Figure 12 shows that the QoS obtained using POC is better than the one using UDP. In fact, when the application receives a loss indication from the partial order transport, it does not wait and anticipates the computing of the following data in the stream, whereas, using UDP, it has to wait because it does not receive loss indication. Thus, the curve presenting the results of the VoD application using UDP grows linearly. The more network losses, the more time wastes, and the more the application has to discard data to ensure intra and inter-streams (between audio and video) synchronisation constraints. The MMPOC loss level is always lower than UDP's one, and the more losses, the more MMPOC is efficient compared to UDP.

6.2 Serial Multi-Networking Results

To illustrate MNP improvements on a serial case, we have been using the following configuration shown in Figure 13.

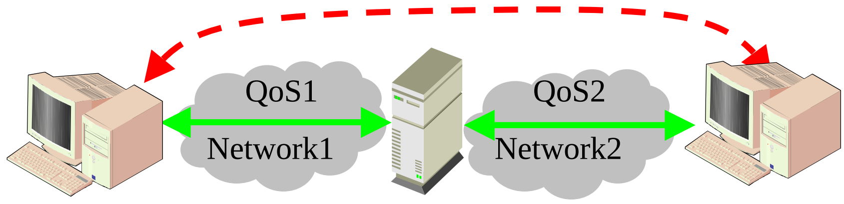


Figure 13 -Serial multi-networking testbed / Exemple de test pour le multi-réseau série

Given this testbed, for each link, we gave the following bit error rate (BER) and end to end delay. The first network (network1) is a WAN relying on optical fibre (BER1=10^{-12} and Delay1=50 ms) and the second one is, for instance, a Wireless LAN (BER2=10^{-6} and delay2=30ms). According to these networks configurations, the end to end delay and the buffer size of the sender have been evaluated. Results were produced from our simulation tools using the MNP model.

It has been shown that the end to end delay is reduced when the global end to end communication is done by cascading MMPOC-MN connections and involving a gateway working as a “dynamic cache”. Results are depicted on Figure 14. It is shown on the same graph the curves corresponding to the following cases: end to end TCP, end to end MMPOC-MN and MNP splitting the end to end connection in two sub-connections and inserting a spoofing gateway in between.

Results show that TCP is not that bad: 60 % of TCP packets have a delay lower than 150 ms, but the tail is very long showing that some packets can experience really long delays. Such delays are very hurtful for all packets and all connections, and they imply to use large buffers. MMPOC with 100 % reliability is quite similar, and it was foreseeable as both protocols are supposed to provide the same service. Differences are just related to the implementation differences between the two protocols. But what is really interesting is that the partial reliability makes performances increase: the delay is much lower as we can discard packets having very long delays and then they do not impact the global performance of the connection.

By splitting the end to end connection, results are even better as all retransmission and acknowledgement mechanisms are optimised for the specific features of each link. As a consequence, delays are very short and even the tail of the curve is very short proving that way that no packet experiences long delay.

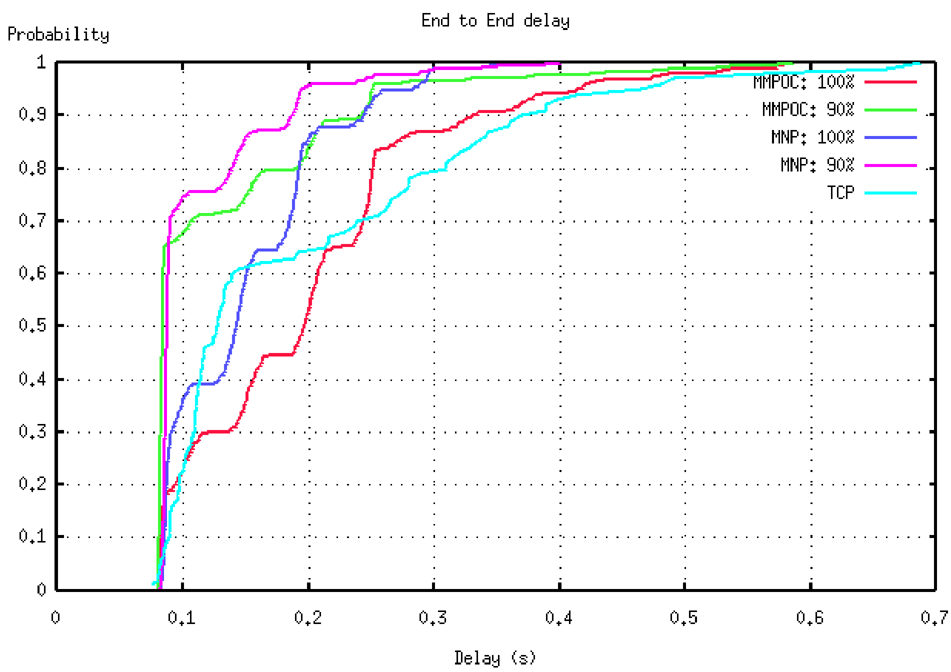


Figure 14 - End to end delay reduction using split consecutive connections / Réduction du délai de bout en bout lorsque les connexions sont partagées

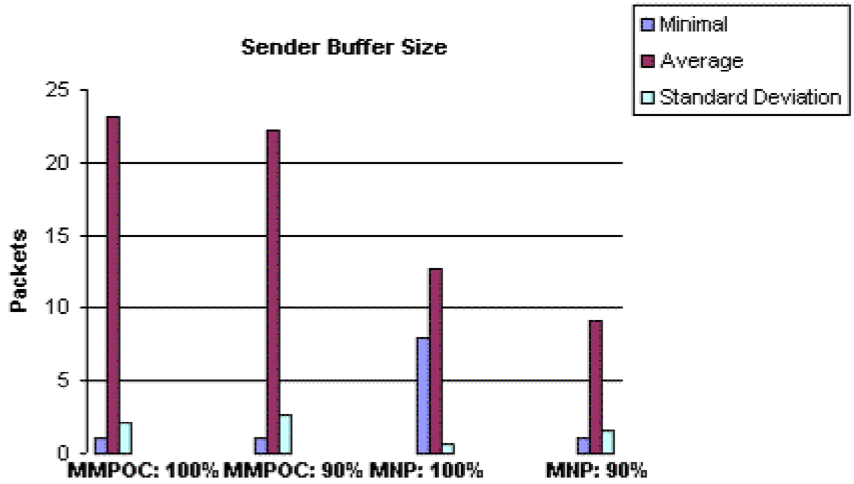


Figure 15 - Buffer reduction using split consecutive connections / Réduction de l'utilisation des tampons avec des connexions partagées

7. Conclusion

This paper presented the two main contributions of this work. First, we tried to classify all applications specific solutions that exist in the Internet or in the literature, and that aim to cope with multi-networking issues. In particular we proposed a different taxonomy, classifying multi-networking systems into serial and parallel ones. And we classified in these two groups either application specific solutions, either generic architecture for enforcing QoS. Both of them can be classified in such a taxonomy.

Based on this taxonomy it appeared that there is no actual way to enforce serial multi-networking for highly interactive applications requiring very low delays. As a follow-up of this observation we proposed a new protocol and architecture for such serial multi-networking called MMPOC-MN and MNP based on two concepts. First, MNP splits transport connections each time packets have to change networks. Second, at the edge of these networks we install kind of a proxy that run spoofing mechanisms to transform each stream profile in the best suited profile for the kind of considered network. This solution also relies on the principles of POC that uses partial order and reliability mechanisms to reduce as much as possible end to end delays. On the other side, MMPOC-MN has been designed for handling parallel multi-networking. For each application, depending on its logical synchronisation model, we can tune our generic protocol to make it perfectly match application requirements and network constraints.

Some performance evaluations showed the benefits of such architecture in both serial and parallel cases. Finally, future work should now be related to this solution deployment and real experiments. It means that we first need to deploy some components in an experimental network able to handle active packets, and then to deploy the MMPOC-MN / MNP solution.

8. References

[6WI01] 6WIND, IP Edge Device, White paper v. 3.0, 2001.

[ALE98] ALEXANDER (D.S.), ARBAUGH (W.A.), HICKS (M.W.), P.K., ANGELOS (D.K.), MOORE (J.T.), GUNTER (C.A.), NETTLES (S.M.), « The SwitchWare Active Network Architecture », *IEEE Network Special Issue on Active and Controllable Networks*, Vol 12 n°.3, pp.29-36, 1998.

[ALL98] ALLMAN (M.), "Ongoing TCP research related to satellite", IETF draft of the TCPSAT Working Group, 1998.

[AME94] AMER (P.D.), CHASSOT (CH.), CONNOLLY (T.), CONRAD (P.), DIAZ (M.), « Partial order transport service for multimedia and other applications », *IEEE/ACM transactions on Networking*, Vol. 2, No. 5,1994.

[BER00] BERTHOU (P.), GAYRAUD (TH.), OWEZARSKI (PH.), DIAZ (M.), "Protocole de communication multimedia multi-réseaux: un nouveau concept", 8th French workshop on protocol engineering (CFIP2000), Toulouse, France, October 17-20 2000, Protocol Engineering, Quality of Service, Multimedia and Mobility, Hermès, ISBN 2-7462-0177-1, 2000.

[BLA98] BLAKE (S.), BLACK (D.), CARLSON (M.), DAVIES (E.), WANG (Z.), WEISS (W.), « An Architecture for Differentiated Services », RFC 2475, December 1998.

[BOR01] BORDER (J.), KOJO (M.), GRINER (J.), MONTENEGRO (G.), "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations", IETF draft of the TCPSAT Working Group PILC: draft-ietf-pilc-pep-07.txt.

[BPK97] BALAKRISHNAN (H.), PADMANABHAN (V.N.), KATZ (R.H.), "The Effects of Asymmetry on TCP Performance," Proc. ACM/IEEE Mobicom, Budapest, Hungary, September 1997.

[CAR96] CARPENTER (B.), "Architectural Principles of the Internet", RFC1958, June 1996.

[DIA95] DIAZ (M.), DRIRA (K.), LOZES (A.), CHASSOT (CH.), "Definition and representation of the quality of service for multimedia systems", *6th International conference on high speed Networking, HPN'95*, Palma de Mallorca, Spain, September 11-15, 1995.

[FOU96] FOURNIER (M.), CHASSOT (CH.), LOZES (A.), DIAZ (M.), "Multimedia partial order transport architecture : Design and implementation. Protocols for High Speed Network", *(PfHSN96)*, 1996.

[HAR96] HARTMAN (J.), MANBER (U.), PETERSON (L.), PROEBSTING (T.), "Liquid Software: A New Paradigm for Networked Systems", *TR96-11*, *University of Arizona Department of Computer Science*, 1996.

[JAV98] JavaSoft, Sun Microsystems, Inc, "HotJava™ Browser : A White Paper", 1998.

[JAV99] JavaSoft, Sun Microsystems, Inc, "Jini™ Technology Architectural Overview". January 1999 (http://www.sun.com/jini/whitepapers/architecture.html).

[INO97] INOUYE (J.), WALPOLE (J.), BINKLEY (J.), "Physical Media Independence : system Support for Dynamically Available Network Interfaces", *OGI CSE Technical Report 97-001*, Janvier 1997. (ftp://www.cse.ogi.edu/pub/tech-reports/1997/97-001.ps.gz)

[KRU98] KRUPCZAK (B.), CALVERT (K.L.), AMMAR (M.), « Implementing Communication Protocols in Java », *IEEE Communications Magazine*, Octobre 1998.

[LAN98] LANDFELDT (B.), SENEVIRATNE (A.), "User Service Assistant(USA): An End to End Reactive QoS Architecture" *6th International Workshop on Quality of Service*, 1998.

[MCC00] McCREARY (S.), CLAFFY (K.), "Trends in wide area IP traffic patterns", ITC seminar on IP traffic measurement, modelling and management, Monterey, CA, September 18th - 20th, 2000.

[OWE98] OWEZARSKI (PH.), DIAZ (M.), CHASSOT (CH.), "A Time Efficient Architecture for Multimedia Applications", *IEEE Journal on Selected Areas in Communication, Special Issue on Protocols and Architectures for Applications of the 21st Century*, 1998.

[PAR97] PARTRIDGE (C.), SHEPARD (T.J.), "Performance of TCP/IP over satellite", IEEE networks, Sept. – Oct. 1997.

[PILC01] Performance Implications of Link Characteristics (pilc). http://www.ietf.org/html.charters/pilc-charter.html

[SCH99] SCHWARTZ (B.), JACKSON (A.W.), STRAYER (W.T.), ZHOU (W.), ROCKWELL (D.), PARTRIDGE (C.), "Smart Packets for Active Networks", *OpenArch*, Mars 1999.

[SRC84] SALTZER (J.H.), REED (D.P.), CLARK (D.D.), "End-To-End Arguments in System Design", ACM TOCS, Vol. 2, No. 4, pp. 277-288, November 1984.

[SRI91] SRINIVASAN (K.), "A control-theoretic approach to flow control.", In SIGCOMM '91 Conference Proceedings, pages 3-15, Zurich, Switzerland, September 3-6, 1991. ACM SIGCOM Computer Communication Review, 21.

[STE98] STEMM (M.), KATZ (R.H.), "Vertical Handoffs in Wireless Overlay Networks", *ACM Mobile Networking (MONET), Special Issue on Mobile Networking in the Internet*, 1998.

[TEN97] TENNENHOUSE ET AL., "A survey Active network Research ", *IEEE Communications Magazine*, vol. 35, No. 1, pp 80-86. January 1997.

[WET98] WETHERALL, GUTTAG, TENNENHOUSE, "ANTS : A Tool kit for Building and Dynamically Deploying Network Protocols ", *IEEE OPENARCH'98*, San Francisco, April 1998.