

Multimedia Partial Order Transport Architecture: Design and Implementation

M. Fournier, C. Chassot, A. Lozes, M. Diaz

LAAS-CNRS

7 avenue du colonel Roche 31077 Toulouse Cédex

email: fournier, chassot, alozes, diaz@laas.fr

URL: <http://www.laas.fr/laas.html>

Abstract

These latest years, technological improvements in Computer and Telecommunication areas led to the emergence of a new kind of distributed multimedia and co-operative applications, whose features and requirements are both multiple and diversified. In order to tackle these needs, two major architectural approaches are currently pursued, one of them arguing for an improvement of the application software, the other one promoting a more sophisticated network support. Following this last approach, this paper aims at providing implementation results performed around the design of a multimedia Transport architecture based on the partial order connection (POC) concept. Design principles of a multimedia partial order Transport connection (MM-POC) are first introduced. Main mechanisms of the associated protocol are then detailed and implementation investigations using the STREAM concept are exposed. Experimentation results are finally given.

Keywords

Multimedia Application, Transport Service, Transport Protocol, Partial Order Connection, Multimedia Transport Architecture, Implementation, STREAM Concept.

1 INTRODUCTION

During the last decade, technological improvements in both Computer and Telecommunication areas led to the emergence of a new kind of distributed *multimedia and co-operative* applications, including transmission and process of all data types (text, fixed image, voice, video sequences, etc.). Requirements of such applications are something new, particularly:

- high speed transmissions are considered to be the first of these needs ;
- temporal constraints have to be enforced when interactive communication is required ; for instance, *real time* transfer is needed to make it acceptable an interactive visioconferencing system ;
- exchanged data have to be orchestrated the ones with the others, this third point raising up what is commonly called a *multimedia synchronization issue* ;
- finally and at the opposite of file data transfers, distributed multimedia applications may tolerate an imperfect transfer of the data they involved, without being unacceptable from users point of view.

Let us note that this list is not complete and may be considered as a subset of all needs present and future applications are going to require. A full analysis of distributed multimedia applications requirements may be found in [30].

Current solutions and their limits

The emergent generation of high speed networks now correctly address the need in high throughputs. Recently, lots of studies have been performed around the design of light weight Transport protocols (such as NETBLT [8], VMTP [4] or XTP [20]) more suited to support multimedia data transfers than TCP or TP4 protocols ; however, these proposals do not provide a sufficient enough solution regard with multimedia requirements complexity. Especially, multimedia synchronization issues (both spatial and temporal ones) are not addressed and temporal constraints management mechanisms are strongly implementation dependent.

A new generation of high speed, multimedia and co-operative protocols has then to be designed on top of high speed links, either starting from the existing solutions, ISDN or Internet protocols, or on top of new ones, ATM-like one. Presently, several communication architectures are being envisaged within different projects that may be classified as follows.

The "Application Aware Networking" approach

Up to now, two major architectural approaches have been proposed to design new distributed communication systems ; both extend the Application Layer Framing (ALF) concept introduced in [9] that argues the interest of Application Data Units (ADU) preservation at the communication system level :

- the first approach assumes that the network software has to be as simple as possible without providing any Quality of Service (QoS) guaranty. The application is supposed to be the best to adapt to network fluctuations by integrating in its software most adequate transport mechanisms. This *Network Aware Application* (NAA) approach is mainly supported by the Internet community, and is currently pursued within the Esprit project HIPPARCH. Recent investigations and results are given in [7] [18] [17].
- conceptually different from the NAA approach, the second approach suggests to increase networking system complexity so as to handle new requirements defined at the applicative level. The major interest of this approach is to make it simpler the user software, at the cost of a more sophisticated network support. This *Application Aware Networking* (AAN) approach has been or is currently pursued within several projects : OSI 95 [11], RACE CIO [19], BERKOM [12], QoS-A [3], TENET [28] and CESAME [15].

Paper Content

Starting from the *partial order connection* (POC) concept [1] [14], work described in this paper follows the AAN approach and is related to the implementation results performed around the proposal of a new multimedia Transport architecture, the *Multimedia Partial Order Connection* (MM-POC), including 'order' and 'reliability' as full QoS parameters [6] [5].

The following of the paper is divided into four major sections. Design principles of a MM-POC are introduced in the first one (section 2) ; their suitability to tackle both synchronization issues and temporal constraints is shown using the Timed Stream Petri Net (TSPN) model [15]. The associated multimedia Transport protocol is detailed in section 3 ; protocol data unit (PDU) structures are presented, then main mechanisms are detailed. Based on the STREAM implementation results concept are developed in section 4 ; it is first shown how the STREAM concept is particularly well suited to implement the AAN approach ; the implementation architecture is then detailed and algorithms are provided. Experimentation results are Finally given and analyzed in section 5.

2 MM-POC: A MULTIMEDIA TRANSPORT ARCHITECTURE

2.1 The Partial Order Connection concept

Currently used Transport protocols are either based on the connection-oriented (CO) paradigm or on the connection-less (CL) one :

- on one hand, CO protocols (TCP like one) provide their users with full reliability and total order, at the cost of increased delay and reduced throughput ;
- on the other hand, CL protocols (UDP like one) introduce no increase in transit delay or reduction in throughput but provide neither order nor reliability guarantees.

This classification makes it clearly appear a design gap between CO and CL protocols that suggests a conceptual link of the classical connection concept. This extension, the *Partial Order Connection* (POC), for which CO and CL connections appear to be special cases, has been introduced and theoretically investigated in [1] and [14]. Basic principles of this recent concept are reminded in the following paragraphs.

A conceptual extension of the connection concept

A POC is an end-to-end connection that allows its users to define and use for transferring data any partially ordered/partially reliable services from no order/no reliability (UDP-like service) to total order/total reliability (TCP-like service). In a POC, 'order' and 'reliability' appear as two specific QoS parameters specified by the service user. In a POC, service data units (SDU) can be delivered to the receiving user in an order different from the sending order : the acceptable difference between the submission sequence and the different delivery sequences precisely results from the definition of the selected partial order.

A suitable concept with regard to multimedia applications features

Aimed at providing a formal representation of temporal constraints in distributed systems, several models have been proposed these latest years, most of them being based on the Petri nets formalism [29] [21] [2].

One of this model, the Timed Stream Petri Net (TSPN) model [16] [26], has been designed to formally

describe multimedia synchronization scenarios in asynchronous distributed systems*. In this model, a place is associated with the presentation of a data (still image or sound fragment for instance) ; logical dependency relationships existing either within a flow or between the different flows of the modeled application are expressed using transitions of the net. The Petri net deduced from a TSPN model provides a logical representation of temporal synchronization constraints, that appears to be a specific partial order.

Consider for instance the *multimedia object* pictured on the left part of figure 1, composed of different *monomedia objects* (a logo, two fixed images and two video sequences) numbered from 1 to 99 ; the picture provides the expected object disposition at the receiving side. The right part of the figure provides the partial order, say P , deduced from the application defined TSPN model, that illustrates (among other) that logo 1 and image 2 may be distributed at the receiving side independently the one with the other ; logical synchronization constraints between the two video sequences are expressed by the intermediate transitions.

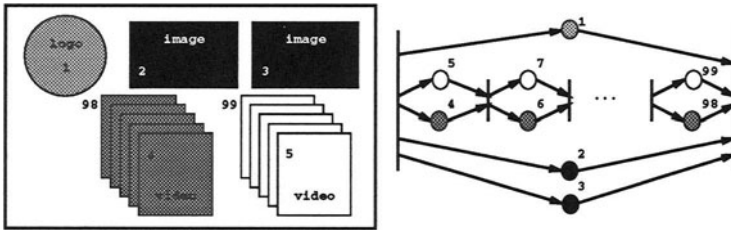


Figure 1 Partial order deduced from a TSPN model of a multimedia object

Provided with a P partial order Transport connection, receiving user may then be delivered monomedia objects (typically service data units) in any sequences consistent with both spatial and temporal synchronization constraints, these different delivery sequences leading to transfer speed-up and save resources at both sending and receiving sides. [1] and [14] illustrate this last point through multiple examples and two different theoretical analysis. Other works performed around the POC concept may be found in [10] and [22].

2.2 MM-POC architecture

From an application point of view, it clearly appears that high speed Transport protocols will be used for multimedia communication. However, as multimedia applications requirements are multiple and diversified with regard to the kind of data flows they imply, it is now necessary to define different transfer characteristics for each of these flows. For instance, assume an application be composed of partially synchronized text and video flows : Transport service has to guaranty both a perfect reliability with respect to the data flow transfer, and a sufficient enough throughput with respect to the video one ; however, a totally reliable Transport connection is not needed for the video flow and a high speed transfer is not a major requirement for a data text-like communication. Moreover, one can mentioned that the use of a reliable service implies transmission latency, often inconsistent with an acceptable high speed video distributed application.

*Other works showing TSPN usefulness in asynchronous operating systems may be found in [23]

Addressing this point, most recent research works led to either an extension of the QoS concept [11] [13] or the proposal of new communication architectures.

As far as this latest point is concerned, two kinds of architecture have been developed :

- the first one provides its users with a given set of service profiles, each of them being able to handle requirements of a specific [28];
- the second one offers a Transport interface whose parameters (throughput, transit delay, jitter) have to be specified and then negotiated for each flow between service users and providers [3] [12]. These works make largely use of investigations around the QoS concept performed in [11].

Although pursuing the AAN approach, both approaches do not tackle multimedia synchronization issues at the Transport level : QoS parameters are defined for each flow, but none of them takes into account dependency relationships between these flows. Our Transport architecture differs from previous ones on this specific point and is based on the use of the TSPN model at different levels of the communication system, particularly at the Transport level.

2.3 Design principles of a MM-POC

In order to take into account the "multi-flow" aspect of a multimedia application, it clearly appears the need for a multimedia Transport architecture providing a set of QoS, each of them being dedicated to the different flows of the application.

In our architecture, a multimedia Transport connection (called a *MultiMedia Partial Order Connection* (MM-POC)) implies the establishment and then a specific co-ordination, at the Transport level, of several *monomedia* connections, each of them providing a distinct QoS.

Consider the MM-POC given figure 2 : three monomedia connections with a given QoS have been established, each of them providing a transport support for a specific data flow (for instance, a video, an audio and a text-like data flow).

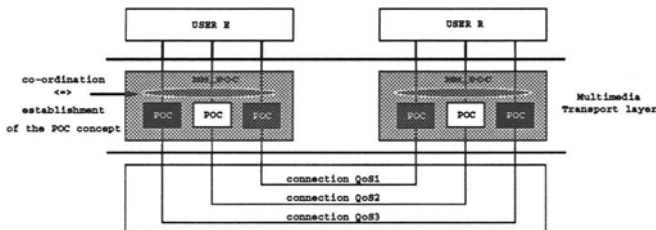


Figure 2 Multimedia Partial Order Transport Architecture

Let us now detail the link between the POC concept and the basic designs stated in the previous paragraph ; in other words, consider how 'order' and 'reliability' parameters can be handled in the proposed architecture.

Order management

Partial order is managed at two different conceptual levels : (1) within each of the monomedia connections (being then specific a monomedia POC) and (2) between these connections. Such a management takes into account at the Transport level both intra- and inter-flow logical synchronization constraints, as they are deduced from the TSPN model.

Consider for instance the multimedia partial order given on the right part of figure 1 , supposed to be deduced from a TSPN model of the object illustrated on the left part. As each color is used to indicate a specific QoS requirements set, the associated MM-POC will then be made of four monomedia POCs, each of them being a Transport support of the same colored objects (typically SDUs).

In this example :

- SDU 6 has to be delivered after SDU 4 is an example of intra-flow dependency relationship; such constraints are managed within the corresponding monomedia POC ;
- SDU 6 has to be delivered after SDU 5 is an example of inter-flow dependency relationship; such constraints have to be managed at a higher conceptual level than the POC one, still within the MM-POC.

It then appears that a multimedia partial order Transport connection allows its users to select the most suitable multimedia partial order service, with regard to temporal synchronization constraints expressed by the application †.

Reliability management

In a monomedia POC, the protocol has not to recover all PDU losses when the resulting lost SDUs do not generate a degradation of the selected reliability. [5] shows how reliability may be managed in two different manners, issuing in each case transit delay improvement for an acceptable reliability decrease.

Both error control mechanisms are based on the following rule :

"delivery of a given SDU makes obsolete all SDUs that are not yet delivered (they can be lost or not) preceding it in the multimedia partial order".

When processing an out of partial order SDU (i.e. not deliverable regard with the multimedia partial order) the protocol is then allowed to deliver it if the number of SDUs made obsolete does not make the maximum loss level exceeded on each POC.

For instance, assume reliability be expressed by the maximum number of consecutive SDU losses, say k_i , the service user may tolerate on POC_i for i from 1 to n (the MM-POC being here supposed to be composed of n monomedia POCs).

Media per media reliability management

When processing delivery of an out of partial order SDU (say A) on POC_i , the protocol will deliver A if PDUs made obsolete on POC_i are still fulfilling reliability requirements ; that is (with our reliability definition assumption) when the number of consecutive valid SDUs (i.e. neither delivered nor lost) made obsolete does not exceed k_i . However, it cannot deliver it if this delivery would need the loss declaration of one or more valid SDUs on any of the other POCs ; a *media per media* reliability is thus defined.

†Due to implementation and real management difficulties, temporal synchronization is supposed to be managed at a higher level than the Transport level.

Per group of media reliability management

Generalizing the previous approach, the second mechanism may now deliver SDU A even if it generates a tolerable number of losses of one or more valid SDUs on any POC, that is (still with our reliability definition assumption) when the number of valid consecutive SDUs made obsolete on POC_j does not exceed k_j for j from 1 to n . This reliability management is said to be *per group of media*.

It then clearly appears that both mechanisms induce transit delay improvement at the cost of an acceptable reliability decrease either on one POC (with a "media per media" error control) or on all POCs (with "per group of media" error control) still enforcing both reliability on each POC and the selected multimedia partial order. Independently of their implementation complexity and the processing time overhead they generate, both mechanisms may be analyzed as follows :

- when a "per group of media" error control is applied, transit delay is optimized on each POC at the cost of a *maximal but acceptable* reliability decrease ;
- differently, a "media per media" error control does not fully benefit from the partial reliability concept, but independence between POCs is preserved.

Let us now consider the implementation aspects of a multimedia partial order Transport connection. The associated MM-POC protocol is first described (section 3).

3 MM-POC PROTOCOL

The MM-POC protocol whose main PDUs and mechanisms are introduced in this section, aims at to ensure continuous media (video or audio-like) Transport support. Two goals have been simultaneously pursued in the protocol design: flow continuity preservation and retransmission delay decrease.

3.1 PDU description

Four different kinds of PDU (see figure 3) have been defined, each of them being identified by a **Type** field:

1. User data are carried over **DATA** type PDUs composed of three sub fields:
 - a **Payload** field whose size may be variable;
 - a sequence number, **Number**, used to identify the carried SDU;
 - an **Ack** field, allowing the sending entity to call for an acknowledgment from the receiving side.
2. **ACK_REQ** type PDU is a short packet also used to request an acknowledgment from the receiving side. No user data can be conveyed within an **ACK_REQ** PDU.
3. **ACK_RES** type PDU is used as a response to an acknowledgment call; it is composed of three sub fields:
 - a **last received** identifier of the last received SDU;
 - a **losses list** list of lost PDUs that have to be transmitted again;
 - the **list size** size of the previous list;

4. **FNACK** type PDU has the same structure as the **ACK_RES** PDU but is sent at the own receiver initiative to provide the sender with the list of PDUs it must transmit again.

Let us now look at the use of these packets within the protocol mechanisms described in the following section.

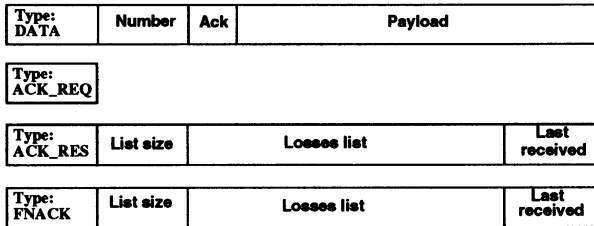


Figure 3 PDU structure

3.2 Protocol mechanisms

Order management

Order is differently managed at the sending and the receiving side. The sending entity is in charge to make a suitable numbering of sent PDUs regard with the application defined multimedia partial order. Receiving a **DATA** PDU, the receiving entity has to verify if the payload (typically a complete SDU) is deliverable with respect to the partial order; undeliverable SDUs are then processed by the reliability management mechanism.

Reliability management

Having to process an undeliverable SDU, reliability management mechanism first checks if the data may be delivered at the cost of “acceptable losses” (see section 2.3). If the data (supposed to be related to POC_i) is not deliverable regard with POC_i reliability QoS, then a **FNACK** PDU is sent by the receiving entity, that makes the sending entity to retransmit unacceptable lost PDU(s). This mechanisms is similar to the one defined in XTP [20].

Two other goals are also simultaneously pursued : flow continuity preservation and retransmission delay decrease. The proposed mechanism is based on a sending window management whose size is a function of a Round Trip Time (*RTT*) measure ; it can be divided into two parts.

1. In order to keep up with the flow continuity and to process data retransmissions as soon as possible, the sending entity periodically asks for an **ACK_REQ** PDU by setting the **Ack** bit every n sent **DATA** PDUs. In order to avoid useless retransmissions, the request period must be greater than the *RTT* value, that makes n to be expressed as follows : $n = RTT * PF$ where *PF* (Production Frequency) indicates the frequency the objects are produced by the sending user.
2. No timer being associated with the previous **DATA** whose **Ack** bit is set up, loss recovery of these

DATA is not ensured: as a result, a close of the sending window may occurred, no free sending buffer being available. In this case, the sending entity has to set up the **Ack** bit of the last sent **DATA** PDU and to manage an associated timer ; while no **ACK_RES** PDU has been received from the pair entity, an **ACK_REQ** PDU is sent each time the timer expires. Timer value has been fixed at $1.1 * RTT$. To make it efficient such a mechanism, the sending window has to be greater than $2 * n$; a simulation study allowed us to conclude a $4 * n$ value was the most suitable regard with the two pursued purposes [24].

4 IMPLEMENTATION

MM-POC implementation choices are described in this section. The implementation architecture is based on the Streams concept [27] whose main principles are now recalled.

4.1 Streams concept

The Streams concept recently appears with the UNIX System V Release 4 norm [27]. This concept allows its users to directly program at the kernel level, inserting or removing *Streams modules*. A module may be seen as a level in a ISO layered structure of protocols [25].

Interest

The AAN approach decreases the applicative software complexity at the cost of a more sophisticated communication software; this implies that the application must use a generic Transport service such as our Partial Order Transport one. The Streams mechanisms permit kernel insertion of optional modules, and is particularly adapted to the AAN approach for the two following reasons:

- First, the layered structure is respected, each level performing its own functionalities using the underlying service.
- Second, Streams modules location within the kernel makes it possible to take advantage of kernel priority in the scheduling.

Streams mechanisms

The notion of stream, or bidirectional (read and write, also called input and output) channel is the major point of the Streams concept. A stream enables service users to dialog with the driver that supplies the service. Communication is done through messages that are exchanged in both directions of the stream.

Additional modules (optional processing units) can be inserted (pushed) then removed (popped) at any time on each stream. They are used to perform additional processing on messages carried on the stream, without having to modify the underlying driver. Within a driver or a module, messages can be either immediately processed by the interface functions that receive them (*put()* routines), or queued in the queues associated with the stream (read and write queues) and then processed asynchronously by *srv()* routines.

4.2 Implementation architecture

Protocol implementation is realized with a Stream module; figure 4 shows the protocol architecture with two monomedia connections. It is important to note that the internal architecture is not symmetrical. This asymmetry is caused by the Partial Order Manager (POM) whose management is only performed at the receiving side. We suppose here that the application sends SDUs in an order matching the selected partial order: it is then not necessary to manage any order at the sending side.

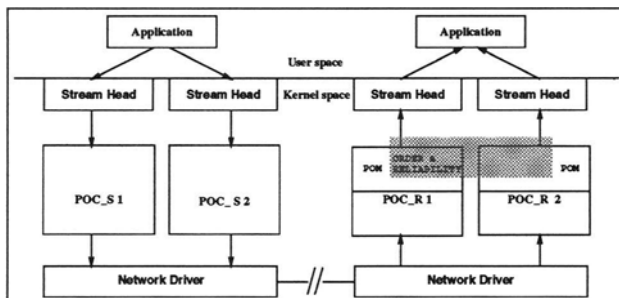


Figure 4 Streams Architecture

At the sending side, each POC_{S_i} is an instance of a generic Stream module in which PDU numbering and PDU retransmission mechanisms are implemented.

At the receiving side, each POC_{R_i} is an instance of a generic Stream module implementing POC_{S_i} mechanisms, but also order and reliability management mechanisms. This latest management, related to the co-ordination of the different POC_{R_i} within the communication software, is performed using a shared data structure illustrating order and reliability multimedia parameters (grey part on figure 4)

The module has been first developed over an IP driver, then on an ATM one.

4.3 Algorithms

Two main algorithms are started up within the protocol, the first one at the sending side, the other one at the receiving side.

Sender algorithm (see figure 5)

Two different kinds of events may occurred at the sending side : a data submission from the sending user (1) or an **ACK_RES** / **FNACK** type PDU receive from the pair entity (2).

1. Having to process a user data, the sending entity performs the following processes :

- **DATA** PDU numbering ;
- storage in the sending buffer ;
- set up of the **Ack** bit if necessary ;
- **DATA** PDU sending ;

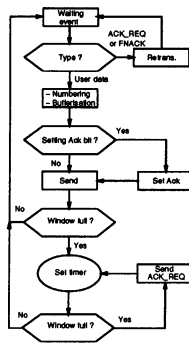


Figure 5 Algorithm at sending side

While the sending window is locked (when the sending buffer is full), a timer is managed at the sending side and an **ACK_REQ** PDU is sent each time the timer expires ; the timer is stopped when an **ACK_RES** or a **FNACK** PDU is received.

2. Receiving an **ACK_RES** or a **FNACK** PDU, the sending entity processes data retransmissions.

Receiver algorithm (see figure 6)

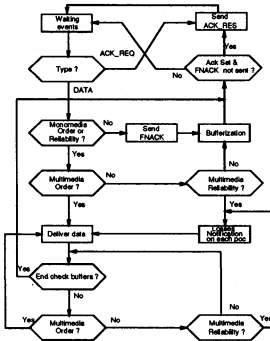


Figure 6 Algorithm of receiving side

Two different kinds of events may occurred at the receiving side : an **ACK_REQ** or a **DATA** type PDU receive from the sending entity.

1. Receiving an **ACK_REQ** PDU, the receiving entity sends an **ACK_RES** PDU indicating the sequence number of the last received **DATA** PDU and the list of **DATA** PDU to be retransmitted.
2. Receiving a **DATA** PDU, the receiving entity performs the following processes:
 - 'Monomedia' order and reliability constraints are first verified:
 - if the received **DATA** PDU does not satisfy both constraints - i.e. when it is not deliverable regard with the monomedia partial order and might be delivered only at the cost of unacceptable losses on the POC it is related to - then a **FNACK** PDU is sent and the **DATA** is stored in the receiving buffers ;
 - else, 'multimedia' order and then reliability constraints are checked:
 - * if the **DATA** cannot be delivered regard with the multimedia partial order, the receiving entity tries to deliver it at the cost of acceptable loss(es) : in this case, the **DATA** is delivered and loss(es) notification is provided to the receiving user through an indication service primitive ; undeliverable **DATA** are stored in the receiving buffers ;
 - * else, the **DATA** is delivered.
 - After each **DATA** delivery, the receiving entity has to check its buffers : stored **DATA** that can now be delivered regard with both monomedia and multimedia constraints are delivered to the receiving user ;
 - If the **Ack** bit of the received **DATA** is set, an **ACK_RES** PDU is sent by the receiving entity only if a **FNACK** PDU has not been sent during the first step of the process.

5 EXPERIMENTATION AND MEASURES

The following study aims at evaluating our MM-POC protocol implementation in a critical network environment that generates both losses and inter-flow desynchronization. In order to enforce multimedia order / reliability QoS parameters, several mechanisms have been designed and two different approaches have been theoretically investigated in section 5.1 and 5.2 : 'media per media' and 'per group of media' ; their impact on the transit delay variation is now evaluated and compared, taking into account the three following parameters: order, reliability and the loss rate generated by the network.

Experimental results provided here have been performed over a multimedia connection ensuring a transport support for an application involving real time distribution of two different continuous media, such as audio and video ones. Measures have been made using our IP/Ethernet LAN with SUN SPARC stations 10. Due to the weakness of the loss level observed on the LAN, artificial losses have been introduced using a dedicated stream module pushed between the driver and each POC. In this module, the 'loss' parameter is expressed by the rate of discarded PDUs over all sent PDUs. In the following measures, this rate is fixed at 10% in order to make it clear the interest of our architecture in a critical environment.

The partial order given figure 7 is supposed to express the logical synchronization constraints deduced from a TSPN modeling a visio-conferencing system.

Objects of the connection 1 are periodically sent all the 40 ms, objects of the connection 2 are periodically sent all the 80 ms.

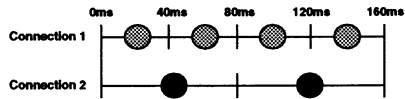


Figure 7 Partial Order

5.1 Media per media management

Experimental results given on figure 8 may be interpreted as follows :

- while the selected reliability (i.e. the acceptable loss number on each POC) is lesser than the network loss rate, retransmissions are performed by the protocol that implies an increase of the average of transit delay variation (ATDV) ;
- when reliability becomes greater than the network loss rate, retransmissions number drastically decreases and the ATDV (around 4 ms on figure 8) may be considered as resulting from the network jitter and the asynchronous feature of the access control CSMA/CD protocol.

5.2 Per group of media management

Compared with results given on figure 8, experimental measures provided figure 9 make appear an analogous global behavior of the two different curves.

Note that when the reliability is greater than 10 %, the ATDV does not present the dispersions observed in the 'media per media' management : this fact is in agreement with the theoretical comparison of both mechanisms that argues for a smaller retransmissions number (at the cost of acceptable loss declarations) when a 'per group of media' mechanism is applied.

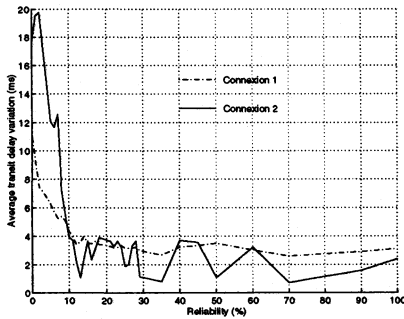


Figure 8 Media per media management

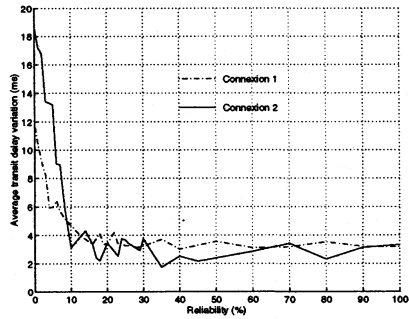


Figure 9 Per group of media management

These results also make appear that a 'per group of media' implementation is no more expensive than a 'media per media' one as far as the processing time is concerned.

In order to point at the 'per group of media' management interest, a second behavior has to be considered : the inter-flow desynchronization. Results and analysis are provided in the next section.

5.3 Management with desynchronization

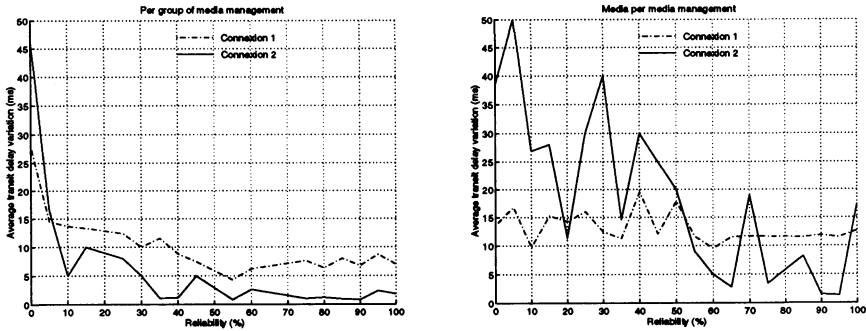


Figure 10 Management with desynchronization

As inter-flow desynchronization can't be significantly observed on our LAN, an artificial 20 ms late has been introduced in each object sending related to connection 2. Experimental results given on figure 10 make it clearly appear a stronger ATDV decrease when a 'per group of media' management is applied.

This phenomena may be easily explicated; when a 'per group of media' policy is applied, the protocol is allowed to deliver data related to flow 1 at the cost of acceptable loss(es) on the late data related to flow 2, this making both flow 1 and flow 2 ATDV be significantly decreased. When a 'media per media' policy is applied, such loss declarations are not allowed and SDUs related to flow 1 have to wait for flow 2 SDU delivery before being delivered.

6 CONCLUSION

This paper first recalled the basic principles of a newly introduced multimedia Transport architecture based on the partially ordered / partially reliable connection concept (POC). This architecture follows the Application Aware Networking (AAN) approach in the design of distributed multimedia applications that argues in favor of a more sophisticated communication software, ensuring a user defined QoS. The major part of the paper consists in implementation results performed around a multimedia partial order Transport protocol whose main mechanisms are introduced here. Particularly, a comparative study of two different multimedia order / reliability management approaches is provided in the paper.

A multimedia partial order Transport connection (MM-POC) implies establishment and then a specific co-ordination, at the Transport level, of different 'monomedia' partial order Transport connections (POC),

each of them providing a suitable QoS with regard to the conveyed flow features. Monomedia POCs co-ordination, that consists in partial order / partial reliability management between POCs, has been envisaged in two different ways : connection per connection (i.e. 'media per media') and per group of connections (i.e. 'per group of media'), both approaches ensuring a QoS guaranty as far as order and reliability parameters are concerned ; a theoretical comparative analysis has been given, arguing that transit delay is improved when a 'per group of media' policy is applied.

Main MM-POC protocol mechanisms and PDUs have been described in the second part of the paper. Designed in order to provide a continuous media Transport support, the MM-POC protocol aims at to ensure flow continuity preservation, fast retransmissions and bandwidth saving, still ensuring both multimedia order and reliability QoS parameters.

In the third part of paper, the MM-POC implementation architecture has been detailed. This architecture is based on the 'Streams' concept that provides a particularly well suited framework to the AAN approach. The integration of both order and reliability management led to the presentation of two efficient algorithms.

Finally, an experimental study has been proposed, implying a typical visioconferencing system traffic on a network generating losses and inter-flow desynchronization. It effectively appeared that a 'per group of media' reliability management was more efficient than a 'media per media' management with respect to the average transit delay variation value.

This experimental work must be followed in two axis: comparison with the Network Aware Application approach and investigation on a WAN using IP networking or ATM native networking with actual losses and desynchronization.

REFERENCES

- [1] P.D. Amer, C. Chassot, C. Connolly, P. Conrad, and M. Diaz. Partial order transport service for multimedia and other applications. *IEEE/ACM Transactions on Networking*, vol.2, n°5, October 1994.
- [2] B. Berthomieu and M. Diaz. Modeling and verification of time dependent systems using time petri nets. *IEEE Transactions on Software Engineering*, 1991.
- [3] A. Campbell, G. Coulson, and D. Hutchison. A quality of service architecture. *ACM Computer Communications Review*, April 1994.
- [4] D.R. Charinton and C.L. Williamson. Vmtp as the transport layer for high-performance distributed systems. *IEEE Communications Magazine*, pages 37-44, June 1989.
- [5] C. Chassot, M. Diaz, and A. Lozes. From the partial order concept to partial order multimedia connections. *Journal for High Speed Networks*, 1996.
- [6] C. Chassot, M. Fournier, A. Lozes, and M. Diaz. Service definition of a multimedia partial order connection. Incs 1052, d. hutchison, h. christiansen, g.coulson, a.danthine (eds.). In *Teleservices and Multimedia Communications, 2nd COST 237 workshop on Teleservices and Multimedia Communications*, Copenhagen, Denmark, November 1995.
- [7] I. Chrismont. Impact of alf on communication subsystems design and performance. In *Proceedings of the 1st Int. Workshop on High Performance Protocol Architectures (HIPPARCH)*, INRIA, Sophia Antipolis, December 1994.
- [8] D.D. Clark, M.L. Lambert, and L. Zhang. Netblt : a bulk data transfer protocol. rfc 998. March 1987.
- [9] D.D. Clark and D.L. Tennenhouse. Architectural considerations for a new generation of protocols. In *Proceedings of ACM SIGCOMM*, 1990.

- [10] P. Conrad, P.Amer, and R.Marasli. Graceful degradation of multimedia documents via partial order and partial reliability transport protocols. In *IEEE Workshop on Multimedia Synchronization*, Virginia, May 1995.
- [11] A. Danthine, editor. *The OSI95 Transport Service with Multimedia Support*. Springer Verlag, 1994. Research Reports ESPRIT, Project 5341, OSI95, Vol.1.
- [12] L. Delgrossi and J. Sandvoss. The berkomi multimedia transport system. August 1993.
- [13] M. Diaz, K. Drira, A. Lozes, and C. Chassot. Definition and representation of the quality of service for multimedia systems. In *6th International Conference on High Speed Networking, HPN'95*, Palma de Mallorca (Balearic Islands), Spain, September 1995.
- [14] M. Diaz, A. Lozes, C. Chassot, and P.D. Amer. Partial order connections. a new concept for high speed and multimedia services and protocols. *Annales des Télécommunications, Tome 49, n° 5-6*, Mai 1994.
- [15] M. Diaz and G. Pays. The cesame project : Formal design of high speed multimedia cooperative systems. *Annales des Télécommunications, Tome 49, n° 5-6*, Mai 1994.
- [16] M. Diaz and P. Sénac. Time stream petri nets for multimedia information. In *LNCS R. Valette (Ed), Int. Conf. on Application and Theory of Petri Nets*, Zaragoza, June 1994.
- [17] C. Diot. Adaptive applications and qos guaranties. In *Invited paper to IEEE Multimedia Networking*, Aizu, Japan, September 1995.
- [18] C. Diot, C. Huitema, and T. Turletti. Network conscious applications. In *HPCS workshop, Mystic*, August 1995.
- [19] L. Henckel. Multimedia communication platform : Specification of the enhanced broadcast transport service. In *CIO RACE Project 2060*, September 1993.
- [20] Protocol Engines Incorporated. January 1992.
- [21] T. Little and A. Ghafoor. Synchronization and storage models for multimedia objects. *IEEE Journal on Selected Areas in Comm., 8(3)*, 413-427, April 1990.
- [22] R. Marasli, P.D. Amer, and P. Conrad. Retransmission-based partially reliable transport service : An analytic model. In *Proceedings IEEE INFOCOM 96*, San Fransico, March 1996.
- [23] P. Owezarski and M. Diaz. Gestion de la synchronisation multimédia dans une application de visioconférence. *Techniques et Sciences Informatiques (TSI)*, 1996.
- [24] L. Robert. Mesure de performances d'un protocole haut débit multimédia de niveau transport. Technical report, LAAS-CNRS, Juin 1995.
- [25] V. Roca and C. Diot. A high performance streams-based architecture for communication subsystems. In *International IFIP Workshop on Protocol for High Speed Networks (PfHSN'94)*, Vancouver (Canada), August 1994.
- [26] P. Sénac, M. Diaz, and P. de Saqui-Sannes. Toward a formal specification of multimedia synchronization scenarios. *Annales des Télécommunications, Tome 49, n° 5-6*, Mai 1994.
- [27] SUN. *STREAMS Programmer's Guide*. 1995. Unix System V Release 4.
- [28] TENET group. *Recent and Current Research*. University of California, Berkeley, USA, April 1994.
- [29] B. Walter. Timed petri-nets for modeling and analysing protocols with time. In *Proceedings of PSTV, III, IFIP*, 1983.
- [30] N. Williams and G.S. Blair. Distributed multimedia applications : a review. *Computer Communications, vol.17, n°2*, February 1994.