Towards a New Generation of Generic Transport Protocols

Patrick Sénac[1,2], Ernesto Exposito[1,2], Michel Diaz[1]

ENSICA, DMI 1 Place Emile Blouin 31056, Toulouse Cedex, France
{Patrick.Senac, Ernesto.Exposito}@ensica.fr
LAAS du CNRS, 7 Avenue du Colonel Roche, 31077, cedex 4, Toulouse, France
Michel.Diaz@laas.fr

Abstract. Considering that current end to end communication services are not adapted for supporting efficiently distributed multimedia application, this paper introduces a new family of generic transport protocols directly instantiated from application layer quality of service requirements. This Generic Transport Protocol (GTP) has been successfully tested for video on demand systems and is one of the major building block of the currently under development GCAP European project. GTP allows one to apply in the transport layer powerful adaptation mechanisms to the network behavior while preserving application requirements and alleviating network bandwidth and buffering needs

1 Introduction

There is a clear need for a new generation transport protocol layer that could apply an efficient adaptation between application layer needs and network behaviour, capabilities and resources. The management of transport connections could be greatly enhanced by informing the transport layer of the reliability, ordering, synchronisation and temporal constraints associated with Application Data Units. Indeed multimedia applications (i.e. video on demand, web browsing, access to MPEG4 or SMIL documents…) do not need the full reliability and total ordering enforced by TCP. Indeed, these applications have partial order, partial reliability and specific synchronization constraints. Therefore, the use of TCP for multimedia applications induces a service that is not only unneeded by the transport service user but above all that can potentially seriously disrupt the semantics of media streams. This reason promoted UDP as the privileged transport layer for accessing to multimedia streams. However, such a solution oblige to introduce into each application complex mechanisms for enforcing application specific data ordering, synchronisation constraints and loss control. Dedicated application layer protocols such as RTP and RTCP do not greatly alleviate the load and complexity of these network aware applications which have to directly adapt their behaviour to the QoS delivered by the network. Therefore, neither UDP nor TCP

TCP are able to offer an efficient service in conformance with the great diversity of application needs. For insuring an efficient mapping between application needs and network behaviour and services, the transport protocol must be aware of the specific ordering, loss and synchronisation constraints related to application data units. Such a generic transport protocol entails an application layer framing approach, which consist in defining, at the application layer, self dependant data units which are also considered by the underlying communication layers as transport data units and network data units [1,2]. If the size of these data units is lower than the size of the maximum transfer unit, such an approach avoids costly fragmentation of data units while allowing the transport protocol to take advantage of data units independency.
TCP and UDP should be two specific instantiations of the considered generic transport protocol that should be able to deliver a continuum of transport services between these two extremities. At the difference of traditional application layer framing approaches which puts all the burden on the application layer and oblige to reinvent the wheel for each application, a generic transport layer has only to be designed once and can be dynamically instantiated to be adapted to specific application needs. In this paper we introduce such a generic transport protocol coupled with a simple and direct derivation technique of a transport layer service from application layer QoS requirements.
In the first part of this paper we briefly introduce a formal technique for modelling multimedia components. Then we demonstrate that this formal approach offers a simple and efficient solution for mapping application layer QoS parameters down to transport layer QoS parameters. Then this formal approach leads us to introduce a new family of Generic Transport Protocol (GTP) that can directly instantiated from the formal expression of application layer QoS requirements. In the last part of this paper we describe a platform independent Java implementation of GTP designed in the framework of the GCAP European Project. Finally two elementary experiments show that, when accessing discrete or continuous media this new generation of transport protocols delivers a service more compliant to the application needs and more efficient than the one offered by UDP or TCP.

2 From Application Layer QoS Requirements to Transport Layer Service.

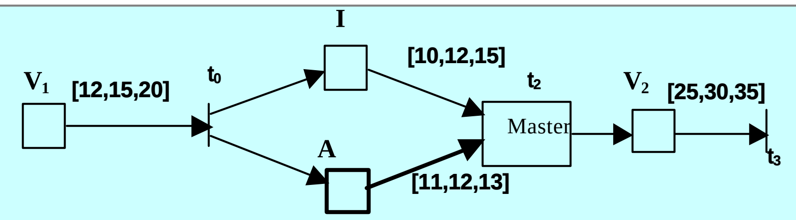
Ideally, a transport protocol should realize efficient adaptations between the greet diversity of application needs and the network behavior. Current transport protocols either, like UDP, deliberately ignore the application needs and the network behavior or like TCP adapt their behavior to network conditions but deliver always a predefined service that ignores specific application needs. For realizing an effective and efficient adaptation between the network and the application, the transport layer, must be informed by the application of its needs. This customization of the transport service can be done when the application creates a transport service access point (e.g. at socket creation).

We propose a three steps approach for using such a generic transport protocol that can apply efficient adaptation decisions between application layer QoS needs and network behavior and services:

1. Definition of application layer QoS needs based on a formal model that allows the consistency of the application requirements to be checked.
2. Formal derivation of a transport service from the application layer requirements
3. Instantiation of the generic transport protocol with the previously obtained transport service.

```
<smil>
 <body>
<seq>
<video src="v1.mpg"  region="region_1" dur=15 min=12s max=20s />
 <par begin="0"  >
   <seq>
      <img src="i.jpg" region="region_1" dur=12s min=10s max=15s />
  </seq>
  <audio src="moliendo.mp3" dur=12s min=11s max=13s syncMaster=true/>
 </par>
<video src="v2.mpg"  region="region_1" dur=30 min=25s max=35s />
</seq>
 </body>
</smil>
```

(b)



(a)

Figure 1. Formal modeling of multimedia components. (a) A SMIL document. (b) The translation of the SMIL document into a HTSPN specification

We have previously shown that the design of complex and large scale distributed hypermedia applications can be greatly enhanced with the help of a formal model that allows the fundamental features of these applications to be specified and their properties to be checked [3,4]. Our approach is based on a temporal extension of Petri Nets, called Hierarchical Time Stream Petri Nets (HTSPN), that allows one to express simultaneously with the same formal techniques the reliability, ordering and temporal constraints associated to a multimedia or hypermedia application. Hence, this formal model allows the most fundamental QoS requirements of multimedia applications to be abstractly expressed. Moreover, taking into account the asynchronous behavior of current network services, this model allows one to specify the admissible temporal variability of multimedia components. The specification of the admissible temporal variability of multimedia components is done with the help of a 3-uple (x,n,y) called ability of multimedia components is done with the help of a 3-uple (x,n,y) called Temporal Validity Interval (TVI), where x, n and y specify respectively the minimal nominal and maximal admissible durations of the component. This model introduces a complete set of synchronization operators which define a formal semantics of synchronization bfor asynchronous or weakly-synchronous systems [5]. This formal semantics suppresses synchronization non-determinism while offering scheduling flexibility for information access, delivery and presentation.

The modeling power of HTSPNs allows the fundamental QoS requirements of advanced hypermedia components, as defined by the soon available SMIL 2.0 standard, to be abstracted and formally expressed (Figure 1).

In summary the HTSPN model allows not only the ordering requirements of the application to be specified through recursive sequential and parallel composition of media elements, but also the reliability and temporal requirements to be expressed with the help of powerful temporal synchronization rules. For instance the specification given in Figure 1-b states that the audio is the master stream of the inter-stream synchronization scheme between the audio stream and the image stream (this is graphically expressed with the help of a bold arrow). This specification is done with the help of a master synchronization transition which states that:

- The audio stream has to be fully presented to the user
- The image stream may be partially presented

3 Deriving a Transport Service from Application Requirements

3.1 The order and Reliability Dimensions

In the previous section we have seen that the HTSPN model allows one to express three fundamental QoS features of multimedia applications:

1. Ordering constraints between the various application data units or components
2. Reliability constraints
3. Time constraints

Because order and reliability constraints are intrinsic to a wide variety of distributed applications this consideration have lead to the definition of two specific transport protocols widely used in the internet, TCP and UDP, which deliver respectively a fully ordered fully reliable service and an not ordered not reliable service. Moreover none of these two protocols take into account application layer temporal constraints. However, as exemplified by the SMIL component in Figure 1, multimedia documents or components neither needs a fully reliable and ordered service nor an unordered and unreliable service. For instance the component modeled in Figure 1 can support the partial or total loss of image I, and the audio and image I can be delivered in any order (i.e. as soon as the transport layer receive the image OR audio component it can be

delivered to the service user). This statement, coupled with the gain obtained from insuring the management of partial order and reliability constraints at the transport level induce a new family of transport protocols which make the most of the application requirements for delivering an optimal service in terms of end to end delay and buffering and bandwidth needs. This new family of “application aware transport protocols” delivers a connection oriented transport service defined from the ordering, reliability and time constraints given by the application when opening a multimedia connection.
Such a generic transport protocol raise the question of a method for deriving the transport layer service from application layer QoS requirements. This mapping between the application layer specification and the transport service can be immediately obtained for the reliability and ordering constraints. Indeed the HTSPN specification defines intrinsically a partial order that is defined as follow.

Definition 1. Let $A=\{a_1,\ldots,a_n\}$, with $I=(1,\ldots n)$, a set of ADUs associated to an application layer QoS specification specified by a HTSPN H. The partial order specification of the transport service related to H is given by the set $O=\{ (a_i)_{i\in p1(I)},\ldots, (a_i)_{i\in pk(I)} \}$ where $P=\{p1,\ldots pk\}$ is a set of permutations on I that defines the partial order on the elements of A directly derived from H.

In other words, a partially ordered transport service can deliver any sequence of ADUs that conforms with the ordering presentation requirements of the application. For example, the partial order constraints of the transport service derived from the HTSPN specification in Figure 1 is given by the set $O=\{(V_1,I,A,V_2),(V_1,A,I,V_2)\}$.

Definition 2. A transport service s that delivers its Transport Service Data Units $A=\{a_1,\ldots,a_n\}$ following the order defined by $o=(a_i)_{i\in p(I)}$ is ordering conformant with the transport service specification S defined by P if and only if $p\in P$.

For instance a transport service that delivers $o==(V_1,I,A,V_2)$ is ordering conformant with the service specification S defined by $O=\{(V_1,I,A,V_2),(V_1,A,I,V_2)\}$.

As seen previously, the synchronization operators introduced by the HTSPN model allow one to distinguish between mandatory application data units an the ones of which the presentation can be partially or totally skipped. More generally the HTSPN model allows a deterministic or probabilistic specification of admissible losses to be expressed. The following definition will consider only the deterministic point of view. The partially reliable transport service associated to an application layer QoS requirement given by a HTSPN H is defined as follows:

Definition 3. Let $A=\{a_1,\ldots,a_n\}$, with $I=(1,\ldots n)$, a set of ADUs associated to an application layer QoS specification specified by a HTSPN H. This HTSPN defines the set $R\subset A$ of ADUs which must be processed by the application. The specification, S, of the partially reliable transport service related to this application layer QoS is also given by the set R which, in this case, defines the set of ADUs that must be delivered to the application by the transport service.

For instance, the partially reliable transport service adapted to the multimedia component modeled in Figure 1 is defined by the set R={V1,A,V2} of the ADUs that must be delivered to the transport service user.

Definition 4. A transport service, that delivers a set r of ADUs, offers a reliability in conformance with a partially reliable transport service specification defined by the set R of mandatory ADUs if and only if $R \subset r$.

By combining definitions 1 and 3 we get the notion of partially ordered and reliable transport service. Such a service which delivers its TSDUs in conformance with both the ordering and reliability constraints defined for the processing of ADUs (i.e. the application layer processing schedule as defined by the formal specification is also considered as a logical access schedule to the transport service) offers the following advantages:

- The application doesn't need to buffer its ADU for reordering purpose
- The application has not to manage reliability constraints
- The receiving transport entity delivers its TSDU as soon as possible in conformance with the application needs.
- Compared to a fully reliable and ordered service, the knowledge of partial ordering and reliability requirements for ADU delivery allows the buffering needs of the receiving and sending transport entities to be reduced.
- Unnecessary retransmission of non mandatory ADUs can be avoided.
- Reliability and ordering constraints introduce some flexibility for ADU transmission schedule. Therefore, in function of the monitored network QoS, the sending transport entities can apply wise filtering or reordering decisions.

Of course, such an application aware transport service has a sound impact on the fundamental transport layer mechanisms such error control, rate control and congestion control.

3.2 The Time Dimension

Before introducing time in transport services and protocols we need first to understand the meaning of "timed transport service". A "timed transport service" can be defined as a service which delivers on time its service data units to the service user. For a multimedia component, composed of several discrete or continuous media, an application data unit must be immediately available when the previous ones have been displayed. For instance, the audio-visual sequence of the multimedia component in Figure 1 (i.e. the inter-stream synchronization scheme between A and I) can finish at any time within the relative interval (i.e. from the beginning of the audio-visual sequence) [11,13].

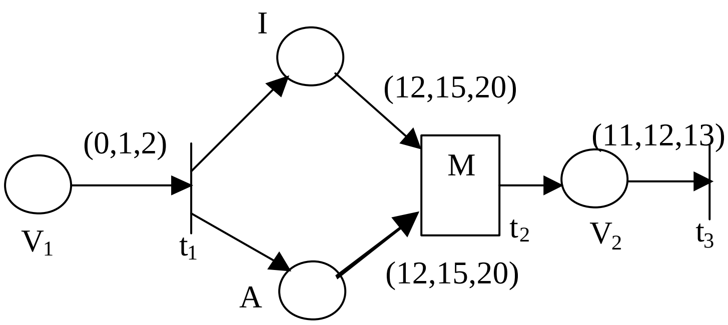


Figure 2. Formal specification of the transport service adapted to the multimedia component modeled in Figure 1. This transport service specification supposes that the service user accepts a 2 time units control time before playing the component.

This time interval is obtained from the synchronization semantics of t_2 transition and from the temporal validity interval of the audio stream which is the master of this inter-stream synchronization point. Considering that this audiovisual sequence can be displayed between 11 and 13 time units the transport protocol deduces that video "V_2" can be delivered to the service user at any time during this relative time interval. Such a definition of a "timed transport service" entails that the service user (i.e. the player of the multimedia component) can adapt the rate of its presentation to the transport service delivery. This limited and accepted adaptive behavior of the application helps the transport protocol to "hide" the variations of the quality of service delivered by the network. Moreover, on time delivery of data to the transport service user spares the application of scheduling access to remote data, of using time-stamping protocols such RTP and of buffering techniques for insuring network jitter and skew reduction. When used on top of a best effort network service this temporal transport service strengthen the isolation between the application and the network and allows wise adaptation and control decision to be applied in consistency with application needs.
Once again such an approach raise the question of defining a temporal transport service adapted to the needs of a given application. The derivation between the application layer time constraints as expressed by a HTSPN model and the related timed transport service is less direct than for the reliability or ordering constraints. This derivation is done by using the following procedure of derivation.

Definition 5. Let us consider an application layer QoS specification modelled by a HTSPN, H. For each ADU, a, in H let us note TVI(a) the Temporal Validity Interval associated to a, and pred(a) the abstract place (as defined in [6]) of H that represents the immediate predecessor of a. The specification of the temporal transport service adapted to H is given by the HTSPN specification H' derived from H as follows:

- $\forall$ a $\in$ A/ pred(a)#$\emptyset$, TVI(a)=TVI(pred(a))
- $\forall$ a$\in$ A / pred(a)=$\emptyset$, TVI(a)=(x,n,y), where x , n and y define respectively the minimum, nominal and maximum admissible waiting time for the delivery of the first ADU

to the transport service user (i.e. the control time that the user accepts for the streaming its multimedia component)

Definition 6. A transport service that delivers a set $A=\{a_1,\ldots,a_n\}$ of ADUs by following the delivery schedule $T=(t(a_1),\ldots,t(a_n))$, where $t(a_i)$ is for the delivery time of a_i, is time conformant with a transport service specification modeled by a HTSPN H' derived from H, if and only if : $\forall$ $a_i \in A/$ $TVI(a_i)=(x_i,n_i,z_i)$ in H', $t(a_i) \in [t(pred(a_i))+x, t(pred(a_i))+y]$. That is, every ADU is delivered in a time window that takes into account the admissible temporal variability of the previous ADUs.

Figure 2 gives the formal modeling of the timed transport service specification associated with the multimedia component modeled in Figure 1. This modeling introduces a control time (a parameter given by the application when opening the transport connection) which specifies the maximum duration accepted by the transport service user before beginning to play the component. Note that the transport service specification can be automatically and simply derived by the transport protocol from the HTSPN that models the application requirements.

3.3 The Space of Temporal Partially Reliable and Ordered Protocols

A multimedia component, as defined in section 2, induces a timed partially ordered and reliable transport service which, in turn, defines a sub-space within the space of the whole family a timed partially ordered and reliable protocols that could be used for transporting the set of application data units which compose this multimedia component (Figure 3) [7].
In this space, a Timed Partially Ordered and reliable transport servic and connection (TPOC) associated to a set of application data units $A=\{a_1,\ldots a_n\}$ is uniquely defined by a 3-uple S=(O,R,T) where:

- O is the set of admissible sequences extracted from A,
- R is the subset of A composed of the elements which must be delivered to the service user,
- TVI is the set of temporal validity intervals associated to the transport service data units delivered by the service S.

For instance for the multimedia component in Figure 1, we have $A=\{V_1,A,I,V_2\}$, and S=(O,R,T) with $O=\{(V_1,A,I,V_2),(V_1,I,A,V_2)\}$, $R=\{V_1,A,V_2\}$ and TVI={(0,1,2),(12,15,20), (12,15,20), (11,12,13)}.

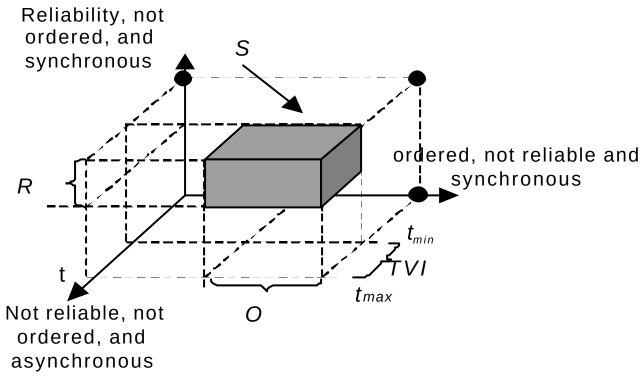


Figure 3. The space of timed partially reliable and ordered transport services

4 Time in Transport Protocols

In the framework of best-effort networks, the transport layer has a fundamental role to play for adapting the network service to application needs. Considering that there is a gap between multimedia applications' temporal requirements and the asynchronous behavior of current networks such as the Internet (i.e. the IP network best effort service), the transport layer is a privileged place where time related QoS parameters can be controlled and enforced. This approach aims to alleviate multimedia applications of the implementation burden of sophisticated buffering and adaptive techniques. Therefore, the design effort of multimedia applications can be greatly reduced by the use of a weakly synchronous transport service (i.e. a TPOC service) which delivers multimedia information units according to time related QoS parameters derived from application level requirements. Such a new generation of transport protocols not only reduces the complexity of distributed multimedia applications, but also entails a dramatic improvement on the use of network and communication resources. Indeed, by taking into consideration at the transport level the temporal semantics of information units, this new approach allows more efficient congestion control, rate control, error control and buffer management techniques to be applied. Indeed, it is well known that the TCP congestion control technique is not adapted to the transport of multimedia streams. The slow start and congestion avoidance mechanisms take uniquely into account the QoS delivered by the network without considering the semantics of the transport service data units. Therefore, with such congestion control mechanisms variations in network QoS impact directly and blindly on the QoS delivered to the transport service user and are instantaneously perceived by the user. Such a behavior is not admissible for continuous media delivery of which the semantic is greatly dependant of time constrains. Our approach allows the transport protocol to react to congestion situations while taking into account the application requirements. This can be done by using the partial reliability and temporal flexibility offered by the concept

of TPOC. Indeed, in function of the QoS delivered by the network, the TPOC can reduce it delivery rate and partially or totally suppress the sending of application data units which can be lost (i.e. the complementary of the set R in U).
Equally, window based rate control mechanisms can be efficiently replaced by rate control mechanisms which take into account the temporal semantics of the transported application data units. Indeed the sending rate can be adjusted dynamically and consistently with the admissible rate variations supported by the transport service, in function of the state of the receiving entity buffer. Note that partial reliability and order can be also used for rate control purpose.
In summary the knowledge by the transport layer of the temporal semantics of the TSDUs offers potentially the following advantages:

- Retransmissions can be more efficiently managed.
- Flow control techniques compatible with application's constraints can be applied.
- Congestion control techniques can be used by combining the flexibility offered by the weakly synchronous time constraints on ADUs.
- The temporal flexibility of ADUs offers multiplexing capabilities and allows network resources to be more efficiently used.
- Associating a temporal duration to ADUs entails a reduction of buffering needs at the receiver side (i.e. the data is received when needed) as well as at the sender side (i.e. by avoiding the buffering of out of date data).
- Avoids the retransmission of out of date data.
- Allows access schedules to ADUs to be managed by the transport layer instead of the application layer.
- Ultimately this approach induces very simple applications which have only to react to transport layer events and unload the management of time and synchronization constraints onto the transport service.

Weakly synchronous transport protocols are also useful in the framework of networks that deliver an integrated or differentiated service. By supporting some temporal admissible variability, these protocols offer some flexibility for network resources management, and offer indirectly to the user and to the network provider, tradeoff facilities between quality and price. A TPOC specification defines an envelope of services, from the worst admissible to best for which the user is ready to pay; these services have to be dynamically adapted and mapped to the network layer differentiated or integrated services. This approach allows the network service provider to satisfy its clients while optimizing its resources usage.

5 GTP Implementation

A first version of a Generic Transport Protocol based on the notion of TPOC has been developed in the framework of the 5th IST European project GCAP (Global Communication Architecture and Protocols for new QoS services over Ipv6 networks).

Table 1. The GTP API

Class GTPServerSocket
This server waits for a connection request from a receiver side
Constructor GTPServerSocket (local Address, local Port, maxConn)
This constructor creates a socket server using a local address, a local port and specifies the maximum number of connections
GTPSenderSocket accept()
This method waits for a connection and accepts it, instantiating a GTPSenderSocket connected to the receiver side

Class GTPSenderSocket
The sender socket is instantiated with the specification of the transport service sent by the peer entity.
Request accept Request()
The accept Request waits for a request from the receiver side
void ackRequet(GTP.Request request)
The ackRequest method acknowledges the requests
void send(GTP.GTPPacket fp)
This method allows one to send GTPpackets to the receiver side

Class GTPReceiverSocket
The receiver socket is able to send the object requests. The "receive" method allows one to read the GTPpackets ready to be delivered to the user
Constructor GTPReceiverSocket(remoteAddress, remotePort, localAddress, localPort)
This constructor creates a GTPReceiverSocket using the local and remote addresses and the ports specified
void closeRequest()
This method allows one to request the termination of the connection
ObjectRequest mediaObjectRequest(GTPMediaObject pmo)
This method instantiates the transport connexion with a specification of service and send a request for a media object to the server side
GTPPacket receive()
The receive method allows one to get a GTPpacket

GCAP aims at developing for the future Internet a new generation of end-to-end multicast and multimedia transport protocols that provide a guaranteed QoS to advanced Multimedia Multipeer applications on top of heterogeneous networks
The Java language has been used for designing GTP because the Java environment offers a multi-platform implementation that delivers the performance required for a transport protocol. Indeed, we have experimented that although C performs better than Java, Java performances are acceptable enough for designing a transport protocol in

the user space and for offering an efficient support to multimedia applications [8]. In its current implemented version GTP offers a generic support to the partial order and reliability constraints required by the transport service user. Temporal services are not yet offered but should be available in the next version.
GTP uses a pull approach, where the receiver side initiates the connection establishment and termination. At the sender side, a server waits for a connection request from the receiver. When the connection request arrives a sender socket is instantiated and connected to the receiver socket. The receiver sends an object request to the sender side in order to get a multimedia component. This object request includes the identification of the multimedia component, and QoS parameters (i.e. a compact representation the TPOC specification of service). Therefore the sending and receiving entities share a common specification of the transport service which allows then to apply efficient adaptation mechanisms between application needs and network behaviour. The GTP API is similar to the TCP standard java API as defined by the socket class of the java.net package (Table 1).

5.1 First Experiment

This first experiment aims to evaluate the gain obtained by using a partial order protocol on top of a non-reliable network environment. The experiment consists in testing the contribution of the GTP protocol for transferring simple JPEG still images between an image server and a remote images player. The considered client server application is able to respectively receive and send independent segments composed of group of macro-block (i.e. these segments can be decoded and displayed in any order). For comparison purpose, this client-server application has been tested successively on top of UDP, GTP, and a fully reliable and ordered instantiation of GTP, hereinafter referred to as TCP*, which aims to simulate the TCP behaviour without its congestion control mechanisms (in its current version GTP does not apply any congestion control technique). Figure 3 illustrates the dummynet based emulation platform used for the experiment. In this experiment, the sender and receiver side are two Windows 2000 systems located in two different subnets. A third computer running the FreeBSD system insures the routing services between the two subnets and supports the network layer emulation environment. Dummynet emulation capabilities have been used for they allow to tune very easily the main network QoS parameter such as bandwidth, losses and delays.

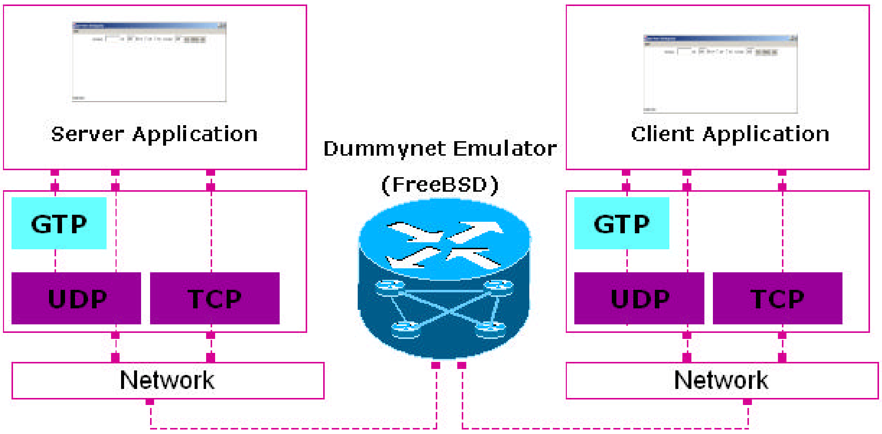


Figure 4. The emulation platform

Figure 5 graphically illustrates the end to end transmission delay required for image 218 Kbytes image. UDP (i.e. no order no loss) , GTP with full reliability and no order, and TCP* (i.e. full reliability and order) protocols are respectively used on top of a network service that induces 0, 5, 10, 15 and 20 percent of losses. The results show that GTP, though instantiated for delivering a fully reliable service, requires almost the same duration than UDP to transmit the data. In contrast, from 10 % of losses, TCP exhibits a dramatic increase of its transmission duration.

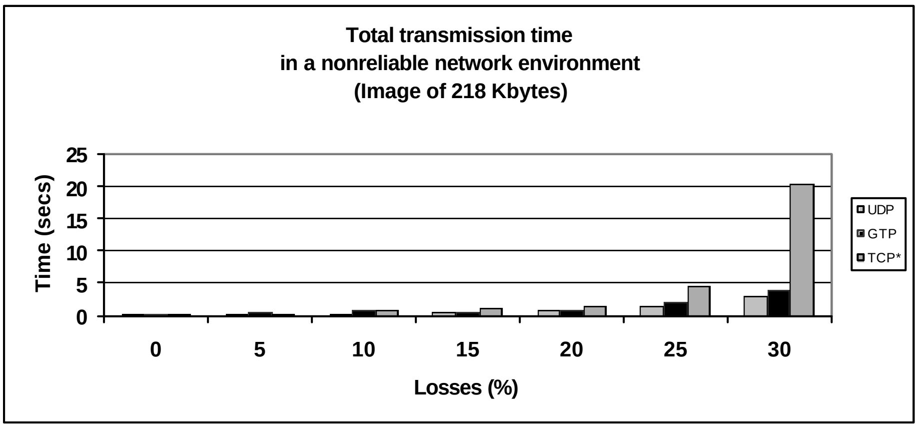


Figure 5. Comparison between UDP, TCP* and GTP end to end transfer delay for a JPEG image

5.2 Second Experiment

The main goal of the second experiment consists in comparing the partially reliable service offered by GTP with the fully reliable service of TCP* (once again TCP is emulated by a fully reliable and ordered GTP connection). The integration capabilities of GTP with existent multimedia applications represent another feature evaluated in this experiment. For this purpose, we have used the Java Media Framework player (JMF Player) for its capabilities to integrate dynamically new transport services and protocols on top of UDP. The same platform that the one described for the experiment 1 has been used for the test bed and for network emulation purpose. The media object to transmit consists in a 1.6 Mbytes MJPEG video stream composed of 411 frames. The partial reliability service of GTP has been instantiated for supporting 30 % of losses. It is important to understand that, in this elementary experiment, because of the fragmentation of TPDU by the network protocol (i.e. video frames are considered as monolithic transport segments), 5% of network losses entails around 35% of losses for the transport layer segments. Note that for avoiding such a segmentation each video frame could be fragmented in independent ADUs like in experiment 1. Such an approach has been described in [9] and [10] where we have proposed a technique that allows MPEG video frames to be segmented into independent ADUs that can benefit not only from a partially reliable transport service but also from a partially ordered one.

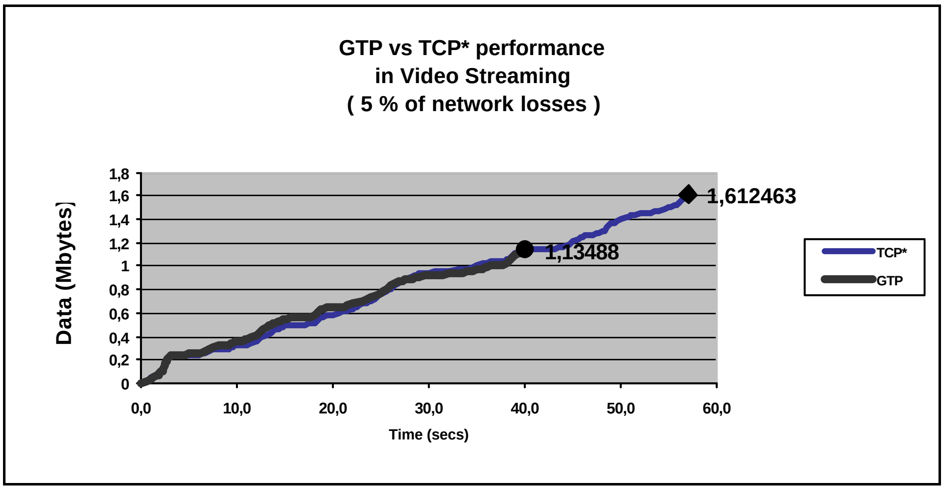


Figure 6. Comparison between a partially reliable GTP instantiation and a fully reliable one (denoted TCP*) for the transport of MPEG video in presence of 5% of network losses.

In presence of 5 % of network losses and with an admissible partial reliability of 30 %, for the GTP service, GTP is able to deliver, within 39 seconds, 70% of the video frames to the transport service user. In contrast a fully reliable service requires 57 seconds for transmitting the full video stream and entail at the application layer long blocking periods that are incompatible with the continuity constraints that must be satisfied for displaying correctly a video stream. Indeed, in this case, the use of a fully ordered

transport service results in 87% of video frames missing their presentation deadlines, to be compared to 47 % with the partially reliable one considered in the experiment.
More generally a partially reliable service offers a trade-off capabilities between the controlled percentage of losses (uncontrolled with an unreliable service such as UDP) and the transmission delay (systematically longer with a reliable transport service such as TCP).

6 Conclusion

On top of highly performing networks there is a clear need of new transport services adapted simultaneously to the new types of services delivered by these networks and to the QOS needs of distributed multimedia applications which are pervading the Internet. We have introduced in this paper a new family of transport services and protocols which satisfy this double requirement. The contribution of this advanced transport services have been successfully experimented for videoconference and video on demand applications [9] and is one of the major building blocks of the currently designed GCAP 5th IST European Program. This new generic transport protocol delivers a service and can exploit rate control, congestion control and error control mechanisms which are directly derived from a formal specification of the application QoS needs. When used on top of best effort networks such an approach enhance dramatically the QoS perceived by the user. This approach is also very promising when used on top of differentiated or integrated services because she offers flexibility for the management of network and end-systems resources, hence GTP allows the user as well the network provider to trade off between performances and cost .
This new field of timed partially reliable and ordered transport protocols offers several open issues. Particularly, the compatibility of the currently studied congestion control mechanisms with TCP connections and TCP receiver or sender entities is a critical research topics which must be solved for insuring the successful dissemination of this new family of protocols. Moreover, extending the point to point approach introduced in this paper for delivering a multicast timed transport service partially reliable and ordered on top of heterogeneous networks is currently experimented in the framework of the GCAP European project.

References

1. Clark D, Tennenhouse D, Architectural Considerations for a New Generation of Protocols, *SIGCOMM* pages 200-208, September 1990.
2. Chrisment I, Huitema C, Evaluating the Impact of ALF on Communication Subsystems, Design and Performance, *Journal of High Speed Networks*, vol. 5, num. 2, Netherlands 1996.

3. Diaz M, Senac P, “Time Stream Petri Nets, a Model for Timed Multimedia Information”, Published in Application and Theory of Petri Nets, Lecture Notes in Computer Science, Springer Verlag Editor, 1994.
4. Senac P., M. Diaz, P. de Saqui-Sannes, A. Leger, "Modeling Logical and Temporal Synchronization in Hypermedia Systems", *IEEE Journal on Selected Areas in Communication*, *special issue on multimedia synchronization*, vol. 14, nº 1, pp. 84 -103, Jan. 1996
5. Courtiat J.P., M. Diaz, R. Cruz, P. Sénac, Formal Models for the Description of Time Behaviors of Multimedia and Hypermedia Distributed Systems, *Computer Communications*, 1996.
6. P. Sénac, P. de Saqui-Sannes, R. Willrich, M. Diaz, “Hierarchical Time Stream Petri Net: A Model for Hypermedia Systems”, Published in Application and Theory of Petri Nets (IEEE), Lecture Notes in Computer Science, Springer Verlag Editor, 1995
7. Amer P., Chassot C., Connolly T., Diaz M., and Conrad P., “Partial order transport service for multimedia applications”, *IEEE/ACM Transactions on Networking*, Vol.2, N°5, pp.440-456, Octobre 1994
8. Apvrille L, Dairaine L, Rojas-cardebas L, Sénac P, Diaz M, “Implementing a User Level Multimedia Transport Protocol in Java”, The Fifth IEEE Symposium on Computers and Communication (ISCC'2000), Antibes-Juan les Pins, France, July 2000
9. L. Rojas, E. Chaput, L. Dairaine, P. Sénac, Michel Diaz, Transport of Video on Partial Order Connections, published in the Journal of Computer Networks and ISDN Systems, 1998.
10. L. Rojas, L. Dairaine, P. Sénac, M. Diaz, Error Recovery Mechanisms Based on Retransmissions for Video Coded with Motion Compensation Techniques, Packet Video 99, New-York, April 99.bb