The QoSxLabel: A Quality of Service Cross-Layer Label

Ernesto Exposito[1], Laurent Dairaine[1,2,3], Mathieu Gineste[2]

[1]National ICT Australia (NICTA), e-mail : firstname.lastname@nicta.com.au
[2]ENSICA, e-mail : firstname.lastname@ensica.fr
[3]LAAS-CNRS, e-mail : firstname.lastname@laas.fr

Abstract— In the context of multimedia and real-time communication, this paper introduces a standardized way for the packet QoS properties to be represented, in order to allow any of the underlying communication mechanisms to access and use these QoS properties. The QoSxLabel (Quality of Service Cross-layer Label) proposes a common syntax expressing the QoS properties. This label is not necessarily added as a new field in the packets but deduced from existing fields according to a well-defined set of rules. The use of the QoSxLabel by some of the mechanisms situated at different levels of the communication architecture will allow a fine optimization of the communication services regarding the real application data requirements.

Index Terms— **QoS, Cross Layer, Label, Multimedia, RTP**

I. Introduction

The development of distributed multimedia applications implies processing and transmission of various multimedia streams. On one hand, communication requirements of such applications include bandwidth and synchronization, and on the other hand, these applications are able to tolerate a non perfect communication service (i.e. a partially ordered and/or partially reliable service). They generally use the concept of Application Level Framing (ALF) [Clark90] as a common design guideline of their communication subsystem. In this approach, Application Data Units (ADU) can be processed independently by the various layered mechanisms to provide the best possible quality of service. The QoS provided is strongly associated to the way this processing has been achieved according to the specific per-flow but also per-packet requirements. Consequently, an essential requirement in order for the mechanisms to achieve an optimized processing is to get an accurate knowledge of ADU Quality of Service properties.

The Real-time Transport Protocol (RTP) [RFC1889] has been proposed as a support for time constrained data communication and follows the ALF principle. RTP is currently largely employed by multimedia applications over the Internet. This protocol integrates information such as timestamps and sequence numbers that can be used by application mechanisms in order to provide the required information to detect and recover losses, reorder data, discard obsolete data and synchronize data flows. Moreover, RTP integrates the data flow identification that enables application to determine the type of media associated with a flow and then to deduce specific QoS requirements (e.g. priorities and dependencies between ADU). Nevertheless, most of RTP applications use UDP or even TCP to transport the ADUs without considering their QoS characteristics. Furthermore, underlying network mechanisms do not usually take into these characteristics. Nevertheless, some network QoS models can make use of flow and packet characteristics, in order to allow packet to provide differentiated services.

Each ADU, produced at application level by a specific codec, convey implicit information that could (and should) be used by the communication subsystem according to its specific properties (e.g., timing, priority, inter or intra dependencies). These properties can be used to optimize most of the communication mechanisms that are involved in the flow communication process, either on the end-hosts, or even on specifics forwarding nodes such as proxies, boundary router or any other intermediate nodes.

In this paper, we introduce a standard representation for the packet QoS properties in order to allow any of the underlying communication mechanisms to access and use the ADU properties. The QoSxLabel (Quality of Service Cross-layer Label) proposes a common syntax expressing the ADU properties. It is deduced from existing ADU fields using a set of well-defined rules. The use of the QoSxLabel by mechanisms situated at different levels of the communication architecture will allow the optimization of the communication processing according to the real ADU requirements.

The rest of the paper is structured as follow. The section II presents the multimedia communication framework considered to deploy the QoSxLabel. In this framework, multimedia communication needs and communication subsystem support for satisfying these needs are proposed. Section III presents the QoS cross layer label (QoSxLabel) and proposes an abstract scheme for its representation and deduction. Section IV describes a study case based on the services provided by the Enhanced Transport Protocol. Finally, some conclusions and perspectives are presented.

II. Multimedia Communication Framework

A. Multimedia Communication Needs

Distributed multimedia applications were first characterized by basic requirements regarding reliability and order of the data being transmitted (i.e. file transfer, web-browsing, etc). As a consequence, the communication services were specifically designed to statically satisfy these basic requirements. An example of this restrictive vision is represented by the widely used transport protocols over IP networks: TCP and UDP.

TCP offers a fully ordered and fully reliable transport service and UDP provides a non ordered and non reliable service. Now, new applications ask for a more complex set of requirements. Applications such as video on demand or videoconferencing on one hand have time, bandwidth and synchronization constraints and on the other hand are able to tolerate a non perfect communication service (i.e. a partially ordered and/or partially reliable service).

In order to satisfy these multimedia applications, new communication services have been proposed. At the transport layer, new protocols have been recently standardized by the IETF. Nevertheless, these new protocols, i.e. SCTP and DCCP, provide only a partial enhancement to the basic transport services already offered by the traditional protocols and fail in taking into account the actual requirements of the applications. In the lower layers, several years of research in network services have led to the specification of new QoS-oriented models such as DiffServ, IntServ, MPLS, etc. However, the complexity involved in the implementation of these network services has limited their widespread deployment. Furthermore, for network and transport services, the remaining problem consists in how to provide the adequate per-packet service aimed at satisfying the application requirements.

B. Application level

The concept of Application Level Framing (ALF), first described by Clark and Tennenhouse [Clark90] is classically applied when designing multimedia systems (e.g. RTP-based applications). The ALF design principle claims for applications to break media data into suitable aggregates. The frames boundaries of these aggregates will be preserved by the lower layers of the communication system. These aggregates are called Application Data Units or ADU and will be used the processing unit. One fundamental contribution of this approach is that each ADU can be processed individually, with respect to other ADU. If the minimum natural size required to preserve ADUs independency is too large to provide a practical unit of transmission, then it will be necessary to define an artificial set of subunits into which an ADU is broken.

A fundamental definition introduced by this principle is represented by the concept of ADU processing representations or “syntaxes”. Peer applications share a common view of the ADU in some “abstract syntax”. The sending side creates a “transfer syntax” in order to describe how the ADU can be used by the receiving side. ALF presents these syntactic forms as a shared namespace in which data elements within the ADU can be identified. Sender and receiver can negotiate the syntax or namespace associated to ADUs in order to use this information to control data transmission. For instance, the syntax describing ADUs composing a video stream could include its sequence number, spatial location and presentation time. This information will be used by receiving application in order to correctly process received ADUs. Moreover, the entities located within the end to end path could also use this information to adjust their operations in order to respect the ADU transmission constraints. To achieve that, this common view should be visible not only for the applications but also for all the communication components present all along the transmission path.

ALF principle provides a pragmatic and efficient approach to share the QoS cross-layer information related to the ADUs between applications and communication system spaces.

C. Transport level

The Transmission Control Protocol (TCP) offers a reliable and in sequence end-to-end data transfer service between two interconnected systems [Postel81]. TCP is a connection oriented and byte-stream oriented service. The User Datagram Protocol (UDP) has been proposed to offer a light transport service for messages or datagrams [Postel80]. UDP is implemented without the time consuming connection phase and without resource consuming error, congestion and rate control mechanisms. In contrast, TCP implements error reporting and recovering mechanisms in order to provide a fully reliable service. Moreover, TCP implements flow and congestion control mechanisms in order to avoid exceeding receiver buffers capacities and network congestion. Error, flow and congestion control mechanisms implemented by TCP may induce transmission delay and variable throughput. In some cases, these effects are not compatible with application requirements, for instance, for multimedia applications demanding guarantees on throughput and delay. For this reason, some of these applications have been implemented using UDP in order to obtain a minimum transport service while deploying ad-hoc mechanisms in order to satisfy their requirements.

The Stream Control Transmission Protocol (SCTP) is a message oriented and reliable transport protocol [RFC2960]. SCTP offers a multi-streams service which means that data can be partitioned in various streams that can be delivered using several independent ordered sequences. Indeed, SCTP does not enforce any ordering constraints between the different streams. It provides a full ordered intra-stream service and a full unordered inter-stream service. This service guarantees that if some loss or disordering is detected in a stream then data delivery over the rest of streams is not affected. In contrast, flow and congestion control are implemented on the association basis and not independently for every stream. The slow-start, congestion avoidance, fast-recovery and fast-retransmission mechanisms are implemented following the TCP algorithms but using the SCTP packets as the acknowledgment unit instead of bytes for the TCP connections.

The Datagram Congestion Control Protocol or DCCP offers a non reliable transport service for datagram flows regulated by a congestion control mechanism [Kohler02]. DCCP is suited to applications currently using UDP. In order to avoid network congestion, applications that use UDP services should implement their own congestion control mechanism. DCCP aims to deliver a transport service that combines both the efficiency of UDP and the congestion control and network friendliness of TCP. Several congestion control mechanisms have been proposed: a TCP-like congestion control using a congestion window, a TCP-friendly rate control or TFRC using an equation to estimate the rate allowed, etc. [Floyd03].

In summary, traditional and new generation of transport protocols have been designed taking into account only a subset of the QoS requirements of multimedia applications.

Indeed, these protocols have been mainly focused to the implementation of congestion control mechanisms to save network resources (i.e. TCP, SCTP and DCCP) while providing full order and full reliability or no order and no reliability at all. In order to optimise the services provided to multimedia applications the Enhanced Transport Protocol (ETP) has been proposed [Exposito03]. ETP services will be described further in this paper.

D. Underlying Network Level

Both at network and at link level, the knowledge of packet characteristics can optimize the processing of the corresponding mechanisms.

The Best-Effort service has been the initial service offered by IP networks and is still the predominant service. It is characterized by an absence of any guarantees in the delivery of data packets. In this model, most of the QoS processing is achieved into the end-systems by the way of transport and application protocols (see above). More recently, services with QoS guarantees managed at network level have been proposed. For example, in the DiffServ (Differentiated Services) approach [RFC2475], boundary routers are processing sophisticated classification, marking, policing and shaping operations depending on per-stream information, while core routers implement simple and fast behaviour aggregate classification. The marking operation in boundary routers is generally based on the Multi-Field (MF) classification consisting in watching multiple fields of packet, such as source address, destination address, TOS byte, protocol ID, source port number, and destination port number. This classification makes the assumption that each user has already been registered to the DiffServ Domain and can be identified by an IP address and port numbers. Another assumption could be that predefined rules for specific protocol ID and port numbers have been set in order to apply particular treatments to these considered streams. Diffserv is an example of approach where in-network mechanisms use cross-layer information. In the DiffServ approach, an intermediate entity, the Bandwidth Broker is sometimes used to help with this cross-layer exchange.

Internet Protocol has been carried over the years over a wide variety of links, with very different characteristics. Among the large set of link layers mechanisms, some of them could provides per-packet treatments functions. For example, in the context of technologies where performances varies a lot, such as Wireless LAN or RF links, it is common that link layers use Automatic Repeat reQuest (ARQ) technique to cope with reliability. In [RFC3366], a classification of the various types of ARQ techniques is proposed. Perfectly persistent ARQ protocols are one that attempts to provide a reliable service. But many arguments exists against the use of such persistence such as the production of uncontrolled delay and jitter in packet delivery, or the fact that application that really need full reliability will implement end-to-end mechanism anyway. Then, high and low persistence link level ARQ protocols have been proposed. The choice between those various ARQ techniques can be based on the knowledge of packet properties for the service to meet the real packet needs.

The same type of questions can be applied to packet scheduling at link layer, which is the subject of a common debate. A scheme that differentiate packet flows into two or more classes, to provide different services at the link layer for each class would be desirable. Then a mean to differentiate the packets and classify them into the various classes is required. Classical ways to achieve this function is to look into a specific field when this classification has been already achieved (e.g., ToS or DSCP in IPv4 header).

Finally, specific link technology can find in these techniques a way to optimize their service. For example, in satellite networking context, the main issue is to make an efficient use of the scarce resources, and more particularly on the satellite return links. Thus, protocols have been designed to optimize carefully the usage of these resources and especially to share properly and efficiently the return link resources accessed by multiple terminals. The use of per-packet specific information could help in this optimization.

III. QoSxLabel: A QoS Descriptor for ADU

The QoSxLabel descriptor is intended to provide information about every ADU composing the different ALF-based multimedia streams. The attributes contained in this descriptor are aimed at helping the different entities participating in end to end path transmission to optimize the real-time operations performed over the ADU. Mechanisms participating in the end to end transmission path could optimize per-packet operations, if enough information about ADU is publicly available. The overall architecture is illustrated in figure 1.

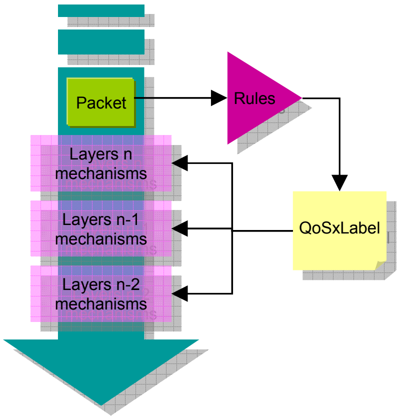

Figure 1. The QoSxLabel Architecture

The label is constituted of well-defined fields which are implicitly present in each ADU. A set of rules depending on ADU coding scheme have also to be provided in order to deduce the QoSxLabel.

A. QoSxLabel Description

The information composing the QoSxLabel includes the following information:

– A unique identification of the ADU, in order to respect reliability and order constraints
– A class of ADU, when within the same media stream different types or classes of ADU are present (e.g. I, P and B pictures of MPEG video streams).
– A priority class (e.g. I pictures are “more important” than P and B pictures in an MPEG stream).
– A tolerated delay, represented by the maximum end to end transmission delay tolerated for the ADU
– An intra-dependency attribute, used for instance when an ADU is too large to be transmitted without segmentation at the communication layers, applications following ALF approach will segment it in several sub-ADUs. In this case, the set of sub-ADUs presents intra-dependency constraints.
– An inter-dependency attribute, for multimedia streams presenting different classes of ADU that can present inter-dependency constraints between the classes (e.g. for MPEG streams, P and B pictures depend on I picture to be decoded).

Next class diagram presents the QoSxLabel specification.

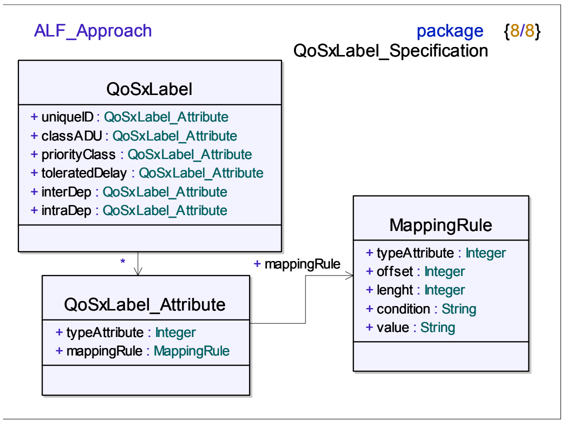
ALF_Approach
package {8/8}
QoSxLabel_Specification
QoSxLabel
+ uniqueID : QoSxLabel_Attribute
+ classADU : QoSxLabel_Attribute
+ priorityClass : QoSxLabel_Attribute
+ toleratedDelay : QoSxLabel_Attribute
+ interDep : QoSxLabel_Attribute
+ intraDep : QoSxLabel_Attribute
*
+ mappingRule
QoSxLabel_Attribute
+ typeAttribute : Integer
+ mappingRule : MappingRule
MappingRule
+ typeAttribute : Integer
+ offset : Integer
+ lenght : Integer
+ condition : String
+ value : String

Figure 2: QoSxLabel specification

B. QoS-ADU mapping rules

In order to allow layered mechanisms to identify or deduce the QoSxLabel descriptor for a particular packet, a set of mapping rules have been proposed. These rules specify the type of attribute (e.g. Integer, Boolean, String, etc), the position of the attribute within the ADU (i.e. offset and length) and optionally conditions to be verified and values to be set if conditions are true.
Further in this paper an example of a QoSxLabel for RTP streams including required mapping rules will be presented.
In order to facilitate the deployment of the QoSxLabel approach, the XML language has been used to define a standard specification including the label attributes and the mapping rules for any multimedia stream. Figure 3 shows an XSD schema document describing the QoSxLabel specification. This XSD schema can be used by codec designers in order to make visible the QoS characteristics of the ADUs.

C. Using the QoSxLabel

Transmission control mechanisms intended to schedule packet transmission and repair errors could use the QoSxLabel descriptor to optimize their operations. Next paragraphs present a non exhaustive list of mechanisms including the possible per-packet attributes required to perform this optimization:
– Flow scheduling: forwarding of packets between end system and networks in an integrated manner. Could be optimized using: tolerated delay, classes and priorities.
– Flow shaping: regulation of flow scheduling based on the flow requirements and also on the available underlying resources. Could be optimized using: tolerated delay.
– Flow policing: actions to be taken when the flow specification is violated (e.g. packet discarding). Could be optimized using: tolerated delay, classes, priorities, inter and intra dependencies.
– Flow synchronization: control of order and time requirements for the delivery of multiple streams (.e.g. audio and video synchronization). Could be optimized using: tolerated delay, inter and intra dependencies.
– Error control: correction of transmission errors (e.g. data corruption or packet losses). Could be optimized using: unique identifier, tolerated delay, classes, priorities, inter and intra dependencies.

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

   <xsd:element name="label">
      <xsd:complexType mixed="false">
         <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:group ref="labelGroup"/>
         </xsd:choice>
         <xsd:attribute name="description" type="xsd:string"/>
      </xsd:complexType>
   </xsd:element>

   <xsd:group name="labelGroup">
      <xsd:choice>
         <xsd:element name="uniqueID" type="labelAttribute"/>
         <xsd:element name="streamClass" type="labelAttribute"/>
         <xsd:element name="ADUClass" type="labelAttribute"/>
               <xsd:element name="classPriority" type="labelAttribute"/>
         <xsd:element name="toleratedDelay" type="labelAttribute"/>
         <xsd:element name="intraDep" type="labelAttribute"/>
         <xsd:element name="interDep" type="labelAttribute"/>
      </xsd:choice>
   </xsd:group>

   <xsd:complexType name="labelAttribute" mixed="true">
      <xsd:attribute name="typeAttribute" type="xsd:integer"/>
      <xsd:attribute name="offset" type="xsd:integer"/>
      <xsd:attribute name="length" type="xsd:integer"/>
      <xsd:attribute name="condition" type="xsd:string"/>
      <xsd:attribute name="value" type="xsd:string"/>
   </xsd:complexType>

</xsd:schema>
```

Figure 3: XSD specification of QoSxLabel and rules

D. Implementation issues

The set of rules and the data packets are sufficient to compute the QoSxLabel partially or totally, where it is needed. A first implementation approach consists in performing the mapping rules in the required devices constituting the end to end communication path. Indeed, this calculation should only be done in the QoSxLabel-aware systems. However, this technique supposes each mechanism that need to access to QoSxLabel information should build its own instance of the QoSxLabel, applying the rules to the current processed packet. This approach is limited by two main problems. First of all, the redundant QoSxLabel construction process can reduce the system performance. Secondly, this approach is not feasible by lower layers mechanisms when encryption is used (e.g., IPSec with ESP payload encryption [RFC2406]).
The first problem could be addressed by sharing the QoSxLabel computation among the mechanisms which makes use of it within the same system (centralized

approach). For instance, the memory space where the label is located can be shared and accessed by the various mechanisms processing the packet. Moreover, to avoid unnecessary computation for QoSxLabel fields that are not going to be used by any mechanisms in the communication architecture, a per-field subscription approach can be implemented.

This technique can be extended to allow any mechanisms to access to these fields in the end to end path in adding an explicit QoSxLabel field into the packet (distributed approach). This technique could also solve the problem of encrypted flows, if the label fields are computed before the encryption process and conveyed in the non encrypted part of the packet.

IV. Study case

In order to evaluate the feasibility of the QoSxLabel implementation as well as the benefits of using this approach in the layered communication mechanisms, an experimental study case including RTP-based applications and transport layer mechanisms has been carried out.

A. RTP-based applications

RTP-based applications offer an ideal framework to deploy the QoSxLabel approach. The following document represents an instance of the XML-based QoSxLabel specification describing the label attributes and the mapping rules for all the RTP streams included in [RFC1889].

```
<?xml version="1.0"?>
<label xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.qosx.org/QoSxLabel.xsd">
   <uniqueID offset="16" length="16"/>
   <intraDep offset="8" length="1"/>
   <toleratedDelay offset="32" length="32"/>
   <streamClass offset="9" length="7"/>
   <!--This general specification is valid for: -->
   <!--A/V streams: GSM(3), G723(4), DVI4-8(5), DVI4-16(6), LPC(7)-->
   <!--    PCMA(8), G722(9), L16-2(10), L16-1(11), QCELP(12), CN(13)-->
   <!--    MPA(14), G728(15), DVI4-11(16), DVI4-22(17), G729(18)-->
   <!--    CelB(25), JPEG(26), H261(31), MP2T(33), Quicktime-F1/F2(96/97)-->
   <!--           And in particular         -->
   <!--video/MPEG MPV,value=32-->
    <ADUClass condition="streamClass==32" offset="117" length="3"/>
    <!--PICTURE-I,value=1-->
    <classPriority condition="streamClass==32.and.ADUClass==1" value="0"/>
    <!--PICTURE-P,value=2-->
    <classPriority condition="streamClass==32.and.ADUClass==2" value="1"/>
    <interDep condition="ADUClass==2" value="0"/>
    <!--PICTURE-B,value=3-->
    <classPriority condition="streamClass==32.and.ADUClass==3" value="2"/>
    <interDep condition="ADUClass==3" value="0,1"/>
   <!--video/H263-A,value=34-->
       <ADUClass condition="streamClass==34" offset="107" length="1"/>
    <!--PICTURE-I,value=0-->
    <classPriority condition="streamClass==34.and.ADUClass==0" value="0"/>
    <!--PICTURE-P,value=1-->
    <classPriority condition="streamClass==34.and.ADUClass==1" value="1"/>
    <interDep condition="ADUClass==1" value="0"/>
</label>
```

Figure 4: XML-based QoSxLabel specification for RTP streams

For all the RTP streams, the *uniqueID*, *intraDep*, *toleratedDelay* and *streamClass* attributes will be found directly in the RTP headers as shown in previous Figure. Specific RTP profiles describing MPEG and H.263 video streams [RFC2250, RFC2190] have been used to map the *ADUClass*, *classPriority* and *interDep* attributes. Next paragraphs presents the advantages of using this QoSxLabel for an ETP transport protocol for transmitting RTP media streams.

B. Enhanced Transport Protocol

The Enhanced Transport Protocol (ETP) is a QoS oriented end to end service aiming to provide a large set of transport mechanisms intended to efficiently satisfy application requirements using available resources and network services [Exposito03]. ETP is message oriented and offers a partially ordered, partially reliable, congestion controlled and timed controlled end-to-end communication service.

QoS Control and management mechanisms provided by ETP operate over the media streams exchanged by applications. These mechanisms require specific QoS information related to the packets composing the media streams in order to be able to provide an optimal service. For instance, a partially reliable service for a video stream needs to know the intra and interdependencies between the packets composing the video pictures in order to assure that all the packets delivered to the application can be decoded. Furthermore, the presentation time of the pictures could also be used in order to implicitly configure this partially reliable service and avoid transmitting and delivering obsolete pictures. For this reason, ETP provides an ideal context to use the QoSxLabel approach.

C. Experiment: Using RTP-QoSxLabel for ETP services

This experiment is intended to demonstrate the importance of taking into account the QoS properties of the packets conveyed by the QoSxLabel in order to optimize the transport services. In this experiment, an implementation of the ETP protocol has been used to transport the video stream of a VoD application, produced at a rate of 133.33 kbps and during a period of 60 seconds. Best-Effort network services, characterized by one way delay of 50 ms and various packet loss rate (PLR), have been emulated. The video profile used for these experiments is H.263, composed by I and P pictures. In this profile, P pictures depend on the previous I picture to be decoded (ADU inter-dependency). Therefore, if an I picture is lost the dependent P pictures cannot be decoded and will be discarded by the receiving application. Furthermore, some I and P pictures are segmented by the application in several packets in order to avoid segmentation at lower layers (i.e. IP segmentation). The various packets composing a single I or P picture present intra-dependency constraints. It means that if any of these packets is lost then the picture will not be able to be completely decoded.

D.Results

Next table presents the comparison between received and useful data for Non-Reliable (NR) and Partially-Reliable (PR) transport services in different network scenarios with packet loss rates (PLR) going from 0% to 90%. At the received side, the actual useful data has been calculated taking into account the intra and inter-dependency constraints. The useful data is smaller that the received data, because some received data has to be discarded by the receiving application when these intra and interdependency constraints have not been respected.

Emulated Network	non reliable service			partially reliable service			
	recv. KB	useful KB	useful/ total	PR	recv. KB	useful KB	useful/ Total
0%	1000	1000	100%	98%	1000	1000	100%
5%	952	905	91%	98%	981	957	96%
10%	892	809	81%	95%	953	886	89%
20%	796	665	67%	90%	902	845	85%
30%	711	571	57%	85%	859	748	75%
50%	521	326	33%	75%	759	609	61%
80%	197	135	14%	60%	593	421	42%
90%	93	34	3%	45%	550	400	40%

These results demonstrate that when no reliability is provided by the transport protocol (i.e. UDP services), part of the received data cannot be used. Furthermore, when a partially reliable service is provided (i.e. ETP services) if the intra an interdependency constraints are not taken into account, the provided service is not in conformance with the real partial reliability required by the application. Moreover, communication resources are wasted transmitting packets which will be discarded at the receiving side.

In order to solve this problem, a Differentiated and Partially Reliable service (D-PR) is provided by ETP. This service can take into account these constraints using the information available in the QoSxLabel. Next table presents the results of a D-PR service for the same experiment. D-PR services have provided a differentiated service for I and P picture of (I,P)={(50%,0%),(100%, 50%)}, respecting the dependency constraints and avoiding wasting network resources.

plr	NR			D-PR: I=50%, P=0%		
		% losses			% losses	
	Time	I	P	time	I	P
0%	60.29	0.00	0.00	60.23	0.00	0.00
5%	60.17	4.50	5.00	60.13	5.50	3.88
10%	60.06	8.00	8.88	60.08	9.50	11.38
20%	60.32	17.50	19.75	60.25	15.00	19.88
30%	60.28	31.50	28.63	60.21	28.00	27.50
	D-PR: I=100%, P=50%			TD-PR		
		% losses			% losses	
plr	time	I	P	time	I	P
0%	60.23	0.00	0.00	60.24	0.00	0.00
5%	60.28	0.00	4.50	60.39	0.00	4.13
10%	60.16	0.00	10.25	60.37	2.00	7.88
20%	60.39	0.00	20.88	60.39	12.00	14.88
30%	60.66	0.00	29.75	60.40	28.50	27.25

However, the delay accumulated for D-PR services depends on the network conditions (i.e. packet loss rates) and sometimes can be incompatible with the application requirements. PR and D-PR services do not take into account these time constraints. In order to permit the adequate reliable service to be provided taking into account time constraints and dynamic network conditions, the Time-constrained, Differentiated and Partially Reliable (TD-PR) service has been evaluated. This service takes into account the maximum tolerated delay available in the QoSxLabel to adjust the provided reliability in order to respect the time constraints. TD-PR service has been configured to provide a D-PR of (I,P)={(100,100),(100,50),(100,0),(50,0)} for a tolerated delay={(*min,target,max*)=(25,150,400)}. The previous table shows that for different network conditions TD-PR optimize the reliability while respecting the time constraints. These results demonstrate that benefits of using the information conveyed by the QoSxLabel in order to optimized the underlying communication services such as the transport services provided by ETP.

V. Conclusion

This paper introduces the QoSxLabel**,** a standardized way for the packet QoS properties to be computed and represented. This label allows any of the underlying communication mechanisms to access and use the QoS packet properties. It is build according to a well-defined set of rules applied on the application data units. The exploitation of the QoSxLabel by several mechanisms situated at different levels of the communication architecture allows the development of optimized communication mechanisms regarding the real ADU requirements. This approach has been successfully implemented and used in the context of the partially ordered, partially reliable, congestion controlled and timed controlled end-to-end communication service provided by ETP. Studies intended to evaluate the performance overhead of the QoSxLabel computation as well as the coherence of the resulting service composition will be carried out.

VI. References

[Clark90] D. D. Clark and D. L. Tennenhouse, "Architectural considerations for a new generation of protocols," in SIGCOMM Symposium on Communications Architectures and Protocols , (Philadelphia, Pennsylvania), pp. 200--208, IEEE, Sept. 1990. Computer Communications Review, Vol. 20(4), Sept. 1990.

[RFC2190] Zhu C., "RTP Payload Format for H.263 Video Streams", RFC 2190, September 1997

[RFC2250] Hoffman D. et al, "RTP Payload Format for MPEG1/MPEG2 Video", IETF Request for Comments: 2250, 1998.

[RFC1889] Schulzrinne H., Casner S., Frederick R., and Jacobson V., "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, Jan 1996

[Postel81] Postel J., "Transmission Control Protocol; DARPA Internet Program Protocol Specification," RFC 793, September 1981

[Postel80] Postel J., "User Datagram Protocol (UDP)", RFC 768, August 1980

[RFC2406] IP Encapsulating Security Payload (ESP)

[RFC2475] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, An Architecture for Differentiated Service, December 1998

[RFC2960] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Scwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, V. Paxson, "Stream Control Transmission Protocol", RFC 2960, October 2000

[RFC3366] G. Fairhurst, L. Wood, Advice to link designers on link Automatic Repeat reQuest (ARQ), RFC 3366, August 2002

[Kohler02] Kohler E. et al, "Datagram Congestion Control Protocol (DCCP)", Internet Draft : draft-kohler-dcp-04.txt, Octobre 2002

[Floyd03] Floyd S. et al, "Profile for DCCP Congestion Control ID 3: TFRC Congestion Control ", IETF Draft, 2003.

[Exposito03] Exposito E. "Specification and implementation of a QoS oriented Transport protocol for multimedia applications". PhD thesis, december 2003.