Building self-optimized communication systems based on applicative cross-layer information

Ernesto Exposito [a,*], Mathieu Gineste [a], Laurent Dairaine [a,b], Christophe Chassot [a]

[a] University of Toulouse, LAAS-CNRS, 7, avenue du colonel Roche, 31077 Toulouse Cedex 4, France
[b] ENSICA, 1, Place Emile Blouin, 31056 Toulouse Cedex 5, France

Keywords:
Self-optimizing
QoS
Cross-layer
Multimedia

A B S T R A C T

This article proposes the Implicit Packet Meta Header (IPMH) as a standard method to compute and represent common QoS properties of the Application Data Units (ADU) of multimedia streams using legacy and proprietary streams' headers (e.g. Real-time Transport Protocol headers). The use of IPMH by mechanisms located at different layers of the communication architecture will allow implementing fine per-packet self-optimization of communication services regarding the actual application requirements. A case study showing how IPMH is used by error control mechanisms in the context of wireless networks is presented in order to demonstrate the feasibility and advantages of this approach.

1. Introduction

Distributed Multimedia Applications have been commonly designed within the framework of the Application Level Framing (ALF) approach [1]. In this approach, multimedia content is fragmented and transmitted as a sequence of Application Data Units (ADUs). The QoS observed by final users is strongly associated to the way these ADUs are transmitted and processed by the communication system. Components participating in the end-to-end communication path are configured to work on per-stream requirements basis (e.g. bandwidth, end-to-end delay, reliability, etc.). However, a more accurate knowledge of individual ADU requirements composing these streams could largely help these components to improve the QoS offered to final users (self-optimization).

An interesting context for deploying this approach is represented by applications based on the Real-time Transport Protocol (RTP) [2]. RTP has become a standard for time constrained multimedia applications following the ALF approach. This protocol defines for every popular multimedia codec, standard ADU headers integrating information such as payload identification, timestamps and sequence numbers. These standard headers are used by receiving applications in order to implement appropriate mechanisms to detect and eventually recover losses, reorder data, discard obsolete data and synchronize multiple streams. RTP also includes additional headers for particular multimedia codecs such as MPEG2, H.263, MPEG4 or H.264 video streams in order to describe their specific properties. Nevertheless, this information is only used at the application layer and is completely ignored by the communication system. Indeed, transport protocols as well as the underlying network and data link mechanisms, do not consider these ADU properties when delivering their services. This information conveyed within the ADU headers could be used by either the end-hosts or specifics forwarding nodes such as proxies or boundary routers to improve the overall QoS [3].

This paper introduces the Implicit Packet Meta Header (IPMH) as a standard interface for publishing the ADU properties of standard or proprietary multimedia streams. This approach is not intended to add new headers to the ADUs but to use existing headers (e.g. RTP headers) to make standard QoS properties publicly available to any underlying communication mechanism. Our contribution also includes a well-defined set of rules aimed at computing the properties to be offered by IPMH for most widely used RTP streams. The use of IPMH by mechanisms located at different levels of the communication architecture will allow illustrating the gains obtained by global cross-layer self-optimization of the communication services.

The rest of the paper is structured as follow. Section 2 introduces the RTP standards proposed by the IETF and suited to develop the IPMH approach. Section 3 presents the multimedia communication framework considered to deploy IPMH. Section 4 presents the standard interface proposed by IPMH and the computation rules for RTP standards.

* Corresponding author.
E-mail addresses: ernesto.exposito@laas.fr (E. Exposito), mathieu.gineste@laas.fr (M. Gineste), laurent.dairaine@ensica.fr (L. Dairaine), christophe.chassot@laas.fr (C. Chassot).

Section 5 describes an illustrative example of the use of IPMH to provide cross-layer QoS self-optimization for error control mechanisms in the context of a wireless network scenario. Concluding remarks, limitations and perspectives of this work are finally proposed.

2. The multimedia transport standard: the Real-time Transport Protocol (RTP)

2.1. Introduction

The Real-time Transport Protocol (RTP) proposed by the IETF [2] has become the standard for time constrained multimedia applications such as video on demand (VoD), audio and video conferencing, voice over IP (VoIP), television over IP (IPTV), etc. RTP has been proposed by the Audio/Video Transport workgroup (AVT) of IETF and an important number of RFCs have been defined in order to describe standards for streaming multimedia content using RTP.

RTP provides end-to-end network transport functions suitable for transmitting real-time data over multicast or unicast network services. RTP is complemented by a control protocol (RTCP) to allow monitoring of the Quality of Service (QoS) and to provide minimal control and identification functionalities.

RTP follows the principles of Application Level Framing (ALF) proposed by Clark and Tennenhouse [1]. ALF principle claims for breaking media data (i.e. audio or video content) into suitable aggregates. The frame boundaries of these aggregates are preserved by lower layers of the communication system (i.e. transport, network and data link-layers). These aggregates are called Application Data Units (ADU) and are intended to be used as the minimal processing unit.

RTP standards follow a header definition approach to describe every ADUs aimed at providing information such as payload identification, sequence numbering and timestamps. RTP, however, does not provide any guarantees concerning the QoS and real-time constraints of the data transported. These guarantees should be provided by lower layers. It's important to note that in a prospect of deriving QoS constraints in communication architectures, the information contained in RTP header is very relevant but currently not taken into account by underlying layers. Next paragraphs introduce the standards describing the common header of the RTP protocol as well as the specific headers used to describe legacy multimedia streams.

2.2. Fixed RTP header

In this section the fields of RTP fixed headers are detailed. These fields need to be specified for every RTP packet composing any RTP media stream (Fig. 1).

- Version (2 bits) identifies the version of RTP (currently version 2).
- P: padding (1 bit) indicates if the packet contains one or more additional padding octets.
- X: extension (1 bit) indicates if the fixed header is followed by exactly one header extension.
- CSRC count (4 bits) contains the number of CSRC or contributing sources for the payload that follow the fixed header.
- M: marker (1 bit) depends of the RTP profile, but generally is intended to specify if the ADU has been segmented in several RTP packets.
- Payload type (7 bits) is used to identify the format of the RTP payload. A set of default format types for audio and video streams is specified in [15].
- Sequence number (16 bits) identifies the order of every RTP packet within the stream.
- Timestamp (32 bits) reflects the sampling instant of the first octet in the RTP data packet.
- SSRC (32 bits) uniquely identifies the synchronization source of the stream (e.g. the sender of a stream of packets derived from a media capture source such as a microphone or a camera).

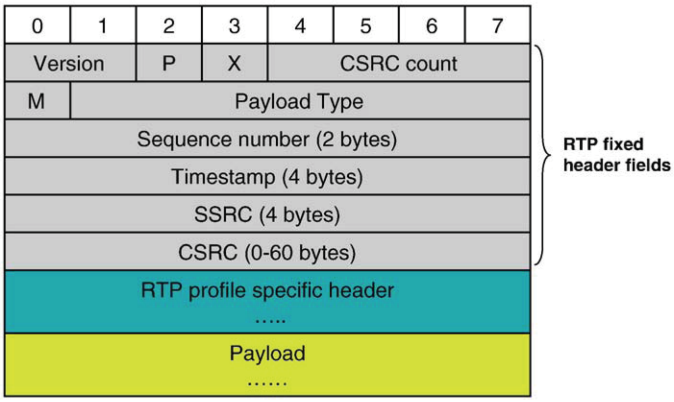

Fig. 1. Fixed RTP header.

- CSRC list (60 bytes) contains the contributing sources for the current payload if it is the result of the combination of multiple streams performed by an intermediate entity.

2.3. Specific RTP profiles

The fixed RTP header can be enhanced with specific headers including additional fields required for more specialized RTP streams such as MPEG1, MPEG2, MPEG4, H.263, H.264, etc.

2.3.1. MPEG

The Moving Picture Experts Group or MPEG, is a working group of ISO/IEC responsible for the development of the widely used video and audio encoding standards such as:

- MPEG-1: standard for multimedia compression mainly used for Video CD and audio MP3.
- MPEG-2: transport and encoding standard for audio and video content mainly targeting Digital Television or DVD.
- MPEG-4: multimedia standard mainly targeting low bitrate encoding for fixed and mobile web and including support for digital rights management. The MPEG-4 part 10 standard includes an advanced video coding also known as H.264 targeting HD DVD.
- MPEG-7: standard for description and search of audio and visual content.
- MPEG-21: defines an open framework for multimedia applications.

In [13] a standard for transporting MPEG1/MPEG2 content over RTP has been proposed. Standards for MPEG-4 transmission over RTP have also been introduced in [17,18], however in this paper only MPEG1/MPEG2 video streams over RTP will be studied. In Fig. 2 the fields of the specific RTP/MPEG video header are presented:

The fields included within this header are intended to be used by the receiving application in order to decode the MPEG video streams. However, some of these fields could be used by the communication system in order to differentiate the priority of RTP/MPEG packets and optimize the utilisation of constrained communication resources:

- P: Picture type (3 bits) indicates the type of picture being transported, 1 for I-pictures, 2 for P-pictures, 3 for B-pictures and 4 for D-pictures. As indicated in [13], I and P pictures could be considered as being more important that B pictures.
- S: Sequence header flag (1 bit) indicates if a sequence-header is present (value of 1). RTP packets containing sequence headers are considered as essential to decode the following packets [13].

Due to space limitations, the rest of the fields contained into the RTP/MPEG headers are not described in this paper. Further information can be found in [13].

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    MBZ  |T|         TR        | |N|S|B|E|  P  | | BFC | | FFC |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                AN              FBV     FFV
```

Fig. 2. Specific RTP/MPEG video header.

2.3.2. H.263

H.263 is a video codec for low-bitrate, originally designed by the ITU-T. In [12] the standard for transporting various versions for H.263 video streams over RTP has been proposed. Fig. 3 presents the RTP/H.263 header for the most widely used H.263 stream.

As for RTP/MPEG streams, some of the fields included within the specific RTP/H263 video stream could be used to optimize the communication resources, in particular:

- P: compression mode (1 bit) indicates the video mode used for video compression. P = 0 implies normal I and P frames and P = 1, indicates I, P and B frames. As for MPEG codec, I pictures are more important than P and B pictures.
- I: intra or inter-coded picture (1 bit) is used to identify the type of picture. I = 0 for intra-coded picture and I = 1 for inter-coded pictures. Inter-coded pictures are dependent of intra-coded pictures.

2.3.3. H.264

The H.264 video codec has a very broad application range that covers all forms of digital compressed video (e.g. low bit-rate Internet streaming applications, HDTV broadcast or Digital Cinema applications) with nearly lossless coding. The codec specification distinguishes between a video coding layer (VCL) and a network abstraction layer (NAL). The VCL contains the signal processing functionality of the codec and a loop filter. The Network Abstraction Layer (NAL) encoder encapsulates the slice output of the VCL encoder into Network Abstraction Layer Units (NAL units), which are suitable for transmission over packet networks. [18] defines a standard way for transporting H.264 content over RTP. The defined RTP payload format allows for packetization of one or more Network Abstraction Layer Units (NALUs), produced by an H.264 video encoder, in each RTP payload. A NAL unit consists of a one-byte header and the payload byte string. The header indicates the type of the NAL unit, the (potential) presence of bit errors or syntax violations in the NAL unit payload, and information regarding the relative importance of the NAL unit for the decoding process. This RTP payload specification is designed to be unaware of the bit string in the NAL unit payload.

Fig. 4 ilustrates the NAL unit header for RTP/H.264 streams.

- NRI: NAL reference identificator (2 bits) a value of 00 indicates that the content of the NAL unit is not used to reconstruct reference pictures for inter picture prediction. Such NAL units can be discarded without risking the integrity of the reference pictures. Values greater than 00, indicate that the decoding of the NAL unit is required to maintain the integrity of the reference pictures. In addition to the specification above, according to this RTP payload specification, values of NRI greater than 00 indicate the relative transport priority, as determined by the encoder. Media Aware Network Elements (MANE), defined in [18] can use this information to protect more important NAL units better than they do for less important NAL units. The highest transport priority is 11, followed by 10, and then by 01; finally, 00 is the lowest. NRI set to 11 generally indicates a coded slice belonging to an IDR (Instantaneous Decoding Refresh) picture. An IDR picture is a coded picture containing only slices with I or SI slice types that causes a “reset” in the decoding process. After the decoding of an IDR picture, all following coded pictures in decoding order can be decoded without inter prediction from any picture decoded prior to the IDR picture.
- Type: NAL unit type (5 bits) specifies the NAL unit payload type as defined in the standard [19]. The type field can be mapped, for instance, to NAL units belonging to an IDR picture or to a non reference picture, and to a sequence parameter set or a picture parameter set.

3. Multimedia communication framework

This section introduces the multimedia communication framework suited to implement and deploy the IPMH approach. Next paragraphs describe the different layers considered in this framework.

3.1. Transport layer

Two main transport protocols are used to transfer information among IP networks. TCP offers a reliable and ordered end-to-end data transfer service between two interconnected systems [4]. TCP is a connection oriented and byte-stream oriented service. It 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 receivers' buffers overflowing and network congestion. UDP was proposed to offer a lightweight transport service with a minimum of protocol mechanisms well suited to time-constrained and multimedia applications (e.g. RTP-based applications) [5].

More recently proposed, the Datagram Congestion Control Protocol (DCCP) offers a non reliable transport service for datagrams, regulated by a congestion control mechanism [6]. DCCP is suited to applications currently using UDP. DCCP aims to deliver a transport service that combines both the efficiency of UDP and the congestion control and network friendliness of TCP. Another protocol recently standardized by the IETF is the Stream Control Transmission Protocol (SCTP) [7]. This is a message-oriented and reliable transport protocol offering multi-stream services. SCTP provides a full ordered intra-stream service and a non ordered inter-stream service. Flow and congestion control are implemented by SCTP following the TCP model but sharing the congestion window between the multi-streams.

In summary, these different transport layer protocols share a common characteristic: none of them takes ADUs' properties into account to optimize the end-to-end QoS.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|F|P|SBIT |EBIT | SRC |I|U|S|A|R      |DBQ| TRB |    TR         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fig. 3. Specific RTP/H.263, 1996-version.

```
+---------------+
|0|1|2|3|4|5|6|7|
+-+-+-+-+-+-+-+-+
|F|NRI|  Type   |
+---------------+
```

Fig. 4. Specific RTP/H.264 header.

3.2. Network layer

The Best-Effort service has been the initial model implemented by IP networks and is still the predominant service of the Internet. It is characterized by the 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 mechanisms (see above). In past years, services offering QoS guarantees at network level have been proposed. For example, Differentiated Services (DiffServ) is a model in which traffic is treated by intermediate systems with relative priorities based on the type of services (ToS) field included in the IP header. The DiffServ architecture defines the DiffServ field (DS), which supersedes the ToS field in IPv4 to make per-hop behaviour decisions about packets processing. In this approach [8], boundary routers process sophisticated classification, marking, policing and shaping operations based on per-stream requirements, while core routers use the previous classification to implement simple, fast and differentiated forwarding. The marking operation in boundary routers is generally based on the Multi-Field classification consisting in inspecting various fields of the packets, such as source address, destination address, protocol ID, source port number, and destination port number in order to set the DS field. This classification makes the assumption that predefined rules exist in order to apply particular treatments. However, the classification granularity is usually the same for all the packets belonging to a given application stream. Indeed, classification and marking processes are only performed taking into account per-stream level information and ignoring the intrinsic ADUs' properties.

3.3. Data-link layer

At lower levels, the IP protocol has been carried over a wide variety of data-link layers, with very different characteristics and QoS. In the context of technologies where performances are subject to important variations, such as wireless networks, it is common that link layers use Automatic Repeat reQuest (ARQ) technique to cope with reliability. In [9], a classification of the various types of ARQ techniques is proposed. Perfectly persistent ARQ protocols provide a fully 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 mechanisms anyway. Then, high and low persistence link level ARQ protocols providing partial reliability have been proposed. Nevertheless, the choices of ARQ techniques are static and technology-dependent and in any case do not use specific ADUs' properties to adapt lower layer services to actual packet requirements.

To summarize, when taken into account, QoS properties of ADUs are only managed at the application layer. Indeed, this information is completely ignored by all the layers of the communication system. The Implicit Packet Meta Header (IPMH) approach is intended to offer a way to discover and use this information in order to allow cross-layer QoS optimization.

4. The Implicit Packet Meta Header

The Implicit Packet Meta Header (IPMH) is intended to make ADUs' properties of legacy multimedia streams publicly available to any layer of the communication system. The attributes available from this standard interface are aimed at helping the different entities participating in the end-to-end path transmission to self-optimize their services by specializing their operations based on the application data requirements and constraints. The overall IPMH architecture is illustrated in Fig. 5.

In this architecture, the IPHM is computed using a specific set of rules deduced from publicly available standards. The IPMH interface can be used by any of the communication components located all along the transmission path, either from the vertical point of view (i.e., cross-layer) or from the horizontal point of view (i.e., set of nodes forwarding the ADU). Obviously IPMH approach can only be used when the application header is publicly accessible to any underlying layer. It means that encryption techniques have not been used at higher layers (e.g. IPsec encryption) and the frame boundaries of the ADUs have been preserved at lower layers (i.e. ALF approach).

4.1. IPMH interface

The IPMH interface is intended to offer read-only access to three categories of attributes: identification, prioritization and dependency attributes:

Identification attributes are generally included as a way of identifying individual ADUs or groups of ADUs belonging to the same stream, as well as to recognize the type of stream and the nature of the multimedia session. The identification attributes category includes:

- Unique ID: uniquely identifying every ADU within the same multimedia stream;
- ADU Type: allowing to identify the various classes of ADU (i.e. substreams) within the multimedia stream (e.g. I, P and B frames for MPEG video streams);
- Stream Type: identifying the nature of the multimedia stream (e.g. audio, video, text, pictures, etc.);
- Session Type: classification of the session based on its requirements (e.g. conversational or interactive, messaging, streaming, gaming, etc.).

Priority-related attributes can be computed for hierarchical media coding streams (e.g. MPEG2, H.263, MPEG4, H.264, etc.). Moreover, when a stream results from the multiplexing of various data sources, the resulting packets might be assigned different priorities in order to differentiate the importance of each of the multiplexed streams. Furthermore, the maximum ADUs' lifetime can also allow deducing their relative priority. The priority-related attributes are:

- ADU Priority: giving the relative priority of a type of ADU (e.g. I pictures are “more important” than P pictures in H.263 or MPEG video streams);

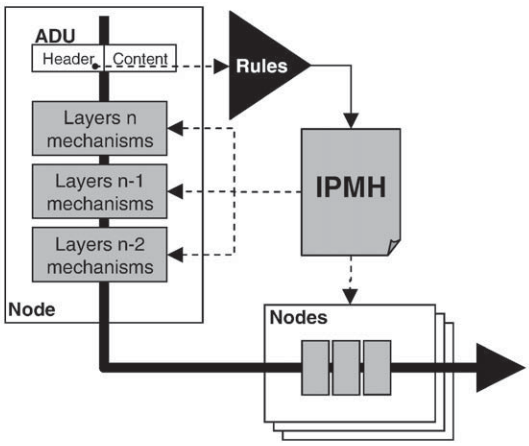

Fig. 5. IPMH architecture.

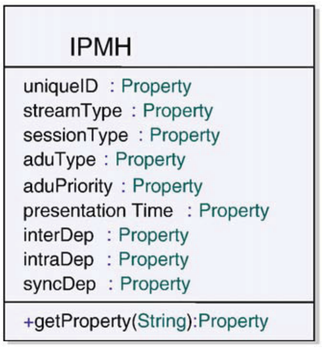

Fig. 6. IPMH interface specification.

- Presentation Time: providing an easy way to estimate the end to end tolerated delay for any given ADU (e.g. 150 ms for interactive applications).

Finally, *dependency-related* attributes includes the dependency relationships that might exist between groups of ADUs in the same stream or between streams belonging to the same multimedia session. For instance, for ADUs exceeding the transmission packet size, dependency relationships exist between the segmented packets composing the ADUs. Likewise, existing compression techniques for hierarchical multimedia coding are generally based on the definition of independent and dependent ADUs in order to reduce the required bandwidth. Dependent ADUs can only be decoded if the reference ADUs are available. These constraints introduce dependency relationships between the ADUs of a stream. Finally, in the case of multimedia sessions composed by more than one stream, synchronization relationships between these streams can also be identified (e.g. video and audio streams). In order to represent these dependency characteristics, IPMH defines the following attributes:

- Intra-Dependency: expressing the dependencies between a set of ADUs representing a segmented application object (e.g. dependency between various segments of an I picture).
- Inter-Dependency: aimed at expressing the dependency relationships between different classes of ADU (e.g. P pictures depend on I pictures)
- Synchronization dependencies: intended to represent the dependencies between synchronized streams of a same applicative session (e.g. lips synchronization between audio and video stream for a videophony session).

The Fig. 6 presents the class diagram of the IPMH specification.

4.2. Self-optimising QoS with IPMH

QoS functions such as packet scheduling or error control could use the IPMH approach to optimize their operations. Next paragraphs present a non exhaustive list of functions and the ADUs' attributes susceptible to be used to perform QoS optimization:

- *Flow scheduling*: forwarding of packets between end-system, networks and sub-networks, buffers and queues management, etc. IPMH could be used to define ADU-level scheduling policies based on the tolerated delay, classes and priorities.
- *Flow shaping*: regulation of flow scheduling based on the flow requirements and underlying resources. This mechanism could use the IPMH in order to limit the accumulated delay to respect the ADU-level tolerated delay.
- *Flow policing*: definition of actions to be taken when the flow specification is violated. Using the IPMH, these actions or policies could be extended in order to respect the ADU-level tolerated delay, classes, priorities and inter and intra dependencies. For instance, the out of profile marking process included in the DiffServ model could be optimized using the classes, priorities and dependencies of the ADUs.
- *Flow synchronization*: control of order and time requirements for the delivery of multiple streams. IPMH could be used to define the synchronization policies in order to take into account the ADU-level tolerated delay and inter and intra dependencies between related streams.
- *Error control*: including detection, reporting and recovery of errors by retransmission or redundancy. The retransmission process could be optimized using tolerated delay, classes, priorities and inter and intra dependencies in order to avoid retransmission of obsolete or less important ADU while respecting inter and intra-dependencies between groups of ADU.

All these functions are performed within the communication system at different levels (i.e. application, transport, network and data-link layers). Due to space restrictions, in this paper only ARQ error control optimization for wireless network will be considered.

4.3. IPMH definition and mapping rules

In order to allow any communication mechanism to obtain a QoS attribute for a particular ADU, a set of mapping rules has to be defined. These rules should specify all the necessary information in order to compute the IPMH attributes: type of attribute to be discovered (e.g. Integer, Boolean, String, etc), position of the attribute within the ADUs' headers (i.e. offset and length) and optionally conditions to be verified. The diagram illustrated in Fig. 7 specifies the mapping rules for the IPMH attributes (properties).

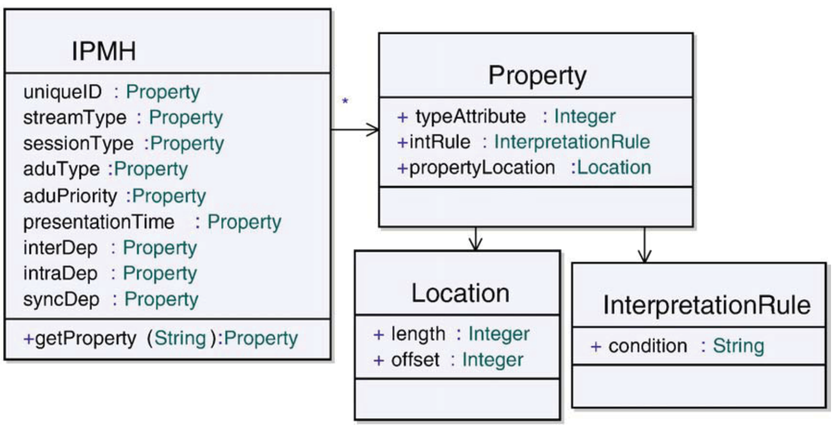

Fig. 7. IPMH mapping rules specification.

4.4. Computing IPMH for standard RTP streams

Next table present the mapping rules required to compute the IPMH attributes for the most common RTP standards defined by the IETF [12–16,18]. For the RTP fixed header a first set of IPMH attributes can be computed directly for any RTP stream. For specific headers (e.g. MPEG, H.263 and H.264), the rest of IPMH attributes can be computed using logic expressions or conditions based on the specific RTP fields.

Stream	Discovered property	Mapping rule
Standard RTP profiles	uniqueID streamType presentationTime intraDependency	[offset,length]=[16,16] [offset,length]=[9,7] [offset,length]=[32,32] [offset,length]=[8,1]
RTP/MPEG (MPV)	aduType aduPriority interDependency	aduType=[offset,length]=[117,[3]] If aduType==1 (I frames) {aduPriority=HIGH interDependency="NONE"} elseIf aduType==2 (P frames) {aduPriority=MEDIUM interDependency="I"} elseIf aduType==3 (B frames) {aduPriority=LOW interDependency="I,P"}
RTP/H.263	aduType aduPriority interDependency	aduType=[offset,length]=[107,[1]] If aduType==0 (I frames) {aduPriority=HIGH interDependency="NONE"} elseIf aduType==1 (P frames) {aduPriority=MEDIUM interDependency="I"}
RTP/H.264	aduType aduPriority interDependency	aduType=[offset,length]=[105,[2]] If aduType==0 {aduPriority=LOW interDependency="NONE"} elseIf aduType==1 {aduPriority=LOW-MEDIUM interDependency="IDR"} elseIf aduType==2 {aduPriority=MEDIUM-HIGH interDependency="IDR"} elseIf aduType==3 {aduPriority=HIGH interDependency="NONE"}

As described in the previous table, for any RTP streams, the *uniqueID, streamType, presentationTime and intraDependency* attributes can be computed from the *fixed* standard RTP header. For specific RTP profiles (i.e. MPEG2, H.263 and H.264 video streams), specific rules have been defined to compute the *aduType, aduPriority and interDependency* attributes.

Likewise, for any proprietary application publicly publishing their ADU headers, the IPMH attributes could also be computed (e.g. Skype VoIP streams).

4.5. Deployment and limitations

IPMH instances and mapping rules need to be publicly available to be used by any communication system layer. Different deployment techniques could be used:

- The installation of IPMH support in a local system or the remote access from a distributed IPMH repository.
- Any IPMH-aware communication mechanism could use an instance of the IPMH interface available from a public IPMH factory that enables to share the same instance of IPMH.

The overhead incurred in discovering a particular property using IPMH is approximately the same that the one incurred when a mechanism performs a standard read operation to get a particular field from the current protocol header (e.g. reading the ToS field of an IPv4 packet). Indeed, this overhead is limited to a function call and optionally one or more conditions evaluation. It is recommended to take into account important issues such as devices capabilities, complexity of interpretation rules and location of the rules evaluator (i.e. server side when receiver devices are limited in processing capabilities), when using and deploying the IPMH approach.

5. Case study: self-optimized error control over lossy wireless networks

In order to evaluate the feasibility of the IPMH implementation as well as the benefits of using this approach in layered communication mechanisms, an experimental case study has been carried out. This case study involves the achievement of various experiments involving an RTP-based Video on Demand (VoD) application and several implementations of ARQ mechanisms. These mechanisms have been designed to use IPMH to discover the ADUs' properties and to self-adapt their error recovery strategies in order to respect the QoS requirements. This study is intended to illustrate how cross-layering self-optimisation between the application layer and an error control mechanisms can be achieved.

5.1. Application layer

The selected VoD application produces a video stream for mobile phones at a rate of 133.33 kbps during a period of 60 s. 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 or P picture to be decoded (ADUs' inter-dependency). It means that if any I or P picture is lost then the dependent P pictures cannot be decoded and will be discarded by the receiving application. Furthermore, I and P pictures can be segmented by the application in several packets in order to avoid segmentation at lower layers. Therefore, if any of these packets is lost then the original picture will not be able to be completely decoded (ADUs' intra-dependency).

5.2. ARQ error recovery strategies

The following strategies have been studied:

- Full reliability (FR): based on the retransmission of every lost packet in order to assure 100% of reliability.
- Partial reliability (PR): intended to provide a partial reliable service by accepting some losses with the objective of reducing the delay induced by retransmissions [10,11]. This mechanism is also implemented using ARQ but the retransmission mechanism is only triggered when the percentage of losses is higher than a specific threshold. For our study, this mechanism takes into account differentiation of ADUs within a same multimedia stream by using the IPMH. This differentiation is achieved using the *classes, inter and intra-dependencies* properties. Two instances of the PR mechanisms for (I,P) pictures have been evaluated:

  PR (50,0) = 50% of I Pictures and 0% of P pictures
  PR (100,50) = 100% of I pictures and 50% of P pictures.

- Time-constrained PR (T-PR): enhances the previous mechanism by taking into account the time constraints of ADUs in order to implicitly configure the error recovery strategy. T-PR strategy aims at providing time QoS guarantees while self-optimizing the reliability perceived by the application. This mechanism is also based in ARQ retransmission, but the threshold of packet losses acceptance is self-configured by the mechanisms regarding the accumulated delay. This mechanism needs to be initially configured by specifying the maximum of packet losses tolerance as well as the maximum of the tolerated delay. The mechanism is able to automatically find the best compromise between delay and reliability by self-adapting the maximum packet loss rate tolerance taking into account the current measured delay. In order to experimentally evaluate this mechanism, two T-PR service configurations are studied:

T-PR (Interactive) = PR for interactive applications going from a minimal PR=(50,0) to a maximal PR=(100,50) while respecting a maximum end to end delay for interactivity of 300 ms.
T-PR (VoD) = PR for VoD applications going from a minimal PR=(50,0) to a maximal PR=(100,50) while respecting a maximum end to end delay for on-demand applications of 10 s.

These error control strategies have been implemented at the transport layer.

5.3. Wireless network scenario

This experiment is intended to illustrate the adaptation properties of the different ARQ strategies using the IPMH approach in the context of a lossy wireless network. A hypothetical wireless network scenario characterized by Packet Loss Rates (PLR) ranging from 0% to 30% and presenting an average RTT of 100 ms has been emulated using a network emulator implemented over a FreeBSD network router. This emulation aims at representing a medium size wireless network presenting various noise, interferences and congestion problems emulated by the PLR and delay parameters.

5.4. Results and analysis

Fig. 8 shows a comparison between the different ARQ strategies for the transmission of the H.263 video stream for the emulated wireless scenario.

The actual loss rate perceived at the application layer is given for the different emulated PLRs. It is important to emphasize that a PLR at the middleware or transport layer can be perceived by the application as a higher PLR (e.g. when a segmented picture cannot be decoded because one of the segmented packets is lost or when a required previous picture is lost). Obviously, the lowest percentage of losses (0%) is guaranteed by the FR strategy. However, as it is illustrated in Fig. 9, this performance in reliability is obtained at the cost of an uncontrolled delay that may be incompatible with the time constraints of the application.

We can also observe that T-PR(VoD) generally conducts to lower loss rates when comparing with the other T-PR or PR mechanisms; this is because T-PR is able to retransmit lost packets when there is enough time to do it. In other words, for VoD applications accepting a higher delay, a higher reliability can be guaranteed by T-PR while respecting a maximum delay of 10 s (Fig. 9.a). Likewise, for interactive applications presenting stronger time-constraints, the highest reliability was provided while respecting a maximum interactive delay of 300 ms (Fig. 9.b).

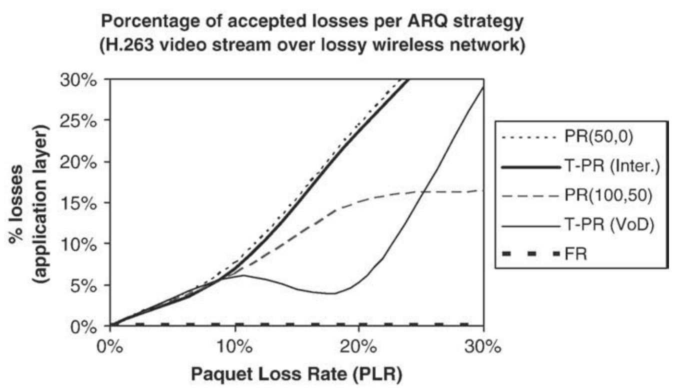

Fig. 8. Losses acceptance comparison of ARQ strategies.

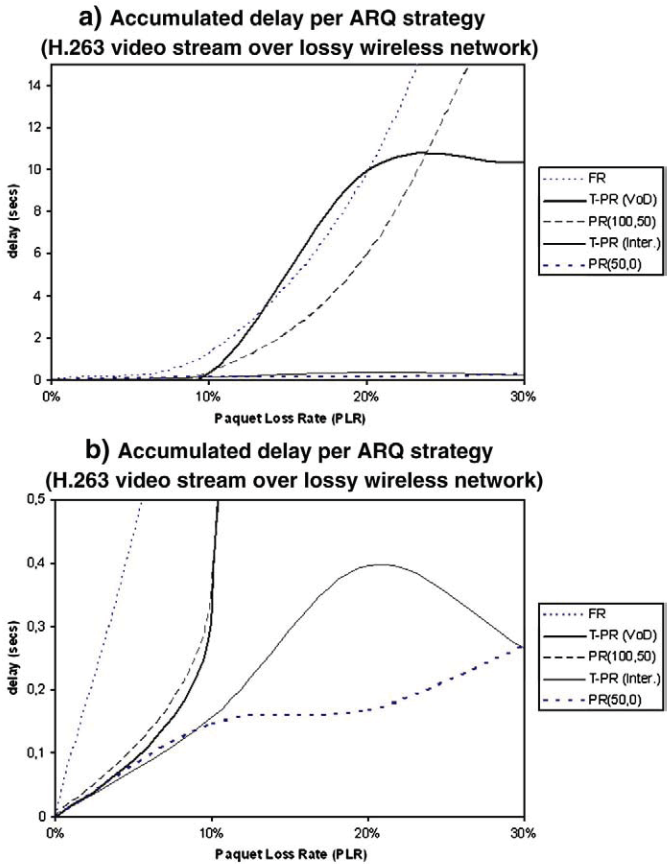

Fig. 9. Accumulated delay comparison of ARQ strategies.

To summarize, for higher reliability a higher delay is usually required. PR and T-PR strategies allow reducing the required delay by accepting some losses. However, loss acceptance can only be achieved if intrinsic ADU constraints are respected; otherwise, loss acceptance could have a higher and uncontrolled impact on the application. IPMH have allowed self-optimize the PR mechanisms by selecting the adequate ADUs to retransmit in order to increase the reliability while respecting ADUs' dependencies. Moreover, T-PR strategies have allowed implementing a self-optimized service by taking into account delay tolerance parameters. Indeed, T-PR is able to optimally select the adequate PR configuration using the time-related information included in the IPMH and regarding the transmission channel conditions.

These results demonstrate how mechanisms implementing ARQ strategies for error control can be self-optimized using the IPMH approach in order to provide the highest reliable service while respecting the ADUs' constraints. This case study illustrates an instance of cross-layer QoS optimisation between application and transport communication layers implementing ARQ mechanisms over lossy wireless networks.

6. Conclusion

This paper introduces the IPMH intended to provide a standard interface of QoS properties for the ADUs composing multimedia streams. IPMH allows underlying communication mechanisms to discover and be aware of the ADUs' QoS properties. The use of the IPMH by these mechanisms allows implementing self-optimized end-to-end communication services regarding the actual ADUs' requirements. This approach has been successfully implemented and evaluated for various ARQ error recovery strategies using the IPMH for optimizing the reliability and delay tolerance properties of multimedia applications. A preliminary set of experiments have illustrated an instance of cross-layer QoS optimisation between applications and transport layers over a lossy wireless networks. Further studies intended to evaluate the overhead added by the computation of the IPMH at lower layer of the communication system (e.g. at network or data-link layers) as well as the distribution and deployment of the IPMH approach are being carried out. These studies will allow to evaluate the viability of using IPMH by other mechanisms such as classification, marking or scheduling packets in the context of QoS networks (e.g. Diffserv or 802.11e models).

References

[1] D.D. Clark, D.L. Tennenhouse, Architectural considerations for a new generation of protocols, Proceedings of IEEE SIGCOMM (Symposium on Communications Architectures and Protocols), Philadelphia, USA, September 1990, pp. 200–208.
[2] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, RTP: a transport protocol for real-time applications, IETF RFC 3550, July 2003.
[3] M. van Der Schaar, Shankar Sai, Cross-layer wireless multimedia transmission: challenges, principles, and new paradigms, IEEE Wireless Communications 12 (4) (2005) 50–58.
[4] J. Postel, Transmission Control Protocol; DARPA Internet Program Protocol Specification, IETF RFC 793, Sept 1981.
[5] J. Postel, User Datagram Protocol (UDP), IETF RFC 768, Aug 1980.
[6] E. Kohler, et al., Datagram Congestion Control Protocol (DCCP), IETF RFC 4340, March 2006.
[7] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Scwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, V. Paxson, Stream Control Transmission Protocol, IETF RFC 2960, October 2000.
[8] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, An Architecture for Differentiated Service, IETF RFC 2475, Dec 1998.
[9] G. Fairhurst, L. Wood, Advice to Link Designers on Link Automatic Repeat reQuest (ARQ), IETF RFC 3366, August 2002.
[10] P. Amer, C. Chassot, C. Connolly, P. Conrad, M. Diaz, Partial Order Transport Service for Multimedia and Other Applications, IEEE ACM Transactions on Networking 2 (5) (October 1994).
[11] E. Exposito, Specification and implementation of a QoS oriented Transport protocol for multimedia applications. PhD thesis, Dec 2003.
[12] J. Ott, et al., RTP Payload Format for ITU-T Rec. H.263 Video, IETF RFC 4629, January 2007.
[13] D. Hoffman, et al., RTP Payload Format for MPEG1/MPEG2 Video, IETF RFC 2250, January 1998.
[14] S. Wenger, et al., RTP Payload Format for H.264 Video, IETF RFC 3984, February 2005.
[15] H. Schulzrinne, RTP Profile for Audio and Video Conferences with Minimal Control, IETF RFC 3551, July 2003.
[16] Y. Kikuchi, et al., RTP Payload Format for MPEG-4 Audio/Visual Streams, IETF RFC 3016, November 2000.
[17] J. van der Meer, et al., RTP Payload Format for Transport of MPEG-4 Elementary Streams, IETF RFC 3640, November 2003.
[18] S. Wenger, T. Stockhammer, M. Hannuksela, M. Westerlund, D. Singer, RTP payload format for H.264 video Internet Engineering Task Force (IETF), RFC 3984, February 2005.
[19] ITU-T Recommendation H.264, Advanced Video Coding for Generic Audiovisual Services, May 2003.

Ernesto Exposito has a lecturer position at the INSA and is a researcher at the LAAS laboratory of the CNRS in Toulouse, France. From 2004, he worked as researcher in the National ICT Australia Limited (NICTA) in Sydney, Australia. He received his engineering degree in Computer Science from the Universidad Centro-Occidental Lisandro Alvarado, Venezuela, in 1995. In 1999, he was awarded a Master of Science in Computer Science from the Paul Sabatier University, France. In 2003 he earned his Ph.D. in "Informatique et Télécommunications" from the Institut National Polytechnique de Toulouse, France. His research interests include: specification, validation, and development of communication services and protocols for multimedia applications in heterogeneous network environments.

Mathieu Gineste is currently a researcher at LAAS-CNRS laboratory (Centre National de la Recherche Scientifique) in Toulouse. He received a Ph.D. in Computer Science in 2006, from Paris 6 University (Pierre et Marie Curie). He did his research during his thesis jointly within the computer science department at ENSICA and the Alcatel Alenia Space research department. He teaches full-year courses related to network computing to INSA and ENSICA "engineering schools" students. He has been involved in European projects dealing with satellite communication enhancements and end-to-end QoS in heterogeneous networks. His research interests concern QoS semantic, QoS management, satellite communication and emulation, and cross-layer enhancements.

Christophe Chassot received the Engineering degree and the M.S. degree (DEA) in Computer Science from the National Polytechnic Institute of Toulouse (INPT) in 1992, and the Ph.D. degree in Computer Science from INPT, in 1995. Since 1996, he has a lecturer position (Matre de Conférences) at the National Institute of Applied Sciences of Toulouse (INSA). He is also an associate Researcher at the Laboratory of Analysis and Architectures of Systems (LAAS), a laboratory of the National Center of the Scientific Research (CNRS). His main fields of interest include QoS-oriented self adaptive Transport architectures, and signaling for multi-domains DiffServ Internet.

Laurent Dairaine is currently an associate Professor at "Ecole Nationale Supérieure d'Ingénieurs de Constructions Aéronautiques" (ENSICA) in Toulouse, France (http://www.ensica.fr). He received a Ph.D. in Computer Science from Pierre et Marie Curie University of Paris in 1994 (LIP6 Laboratory). In 1994 and 1995, he spent a year as visiting researcher at University of technology, Sydney (UTS), Australia. His research interests concern quality of service and end-to-end communication architectures. He recently worked on application and transport protocols for IP Multicast over satellite, quality of service support over peer to peer and replication systems, and network emulation.