Abstract: Development of new communication services demands the complete understanding of the Quality of Service (QoS) requirements and efficient interaction with resources and the other services of the communication system. This paper proposes the use of a global QoS framework based on a common semantic namespace for the deployment of QoS oriented communication services offered by the Fully Programmable Transport Protocol. Experimental results demonstrate the feasibility and advantages of this proposal.

Keywords: Quality of service, transport services, language specification, communication architecture, congestion control.

I. INTRODUCTION

Quality of services (QoS) can be defined as “the set of those quantitative and qualitative characteristics of a distributed multimedia system, which are necessary in order to achieve the required functionality of an application” [15]. At the beginning of Internet, the first communication services were specifically oriented to traditional applications requirements. These requirements were characterized by basic QoS parameters (i.e. either full order and full reliability or no order and no reliability). During last years, new applications requirements have appeared and new QoS oriented communication services have been proposed. But the correct mapping and configuration of these services in accordance to the applications requirements have become a problem not trivial to solve. In this paper we introduce a QoS framework architecture called XQoS. XQoS offers a XML-based semantic namespace for QoS specification. This semantic namespace includes a language specification intended to describe the QoS requirements of users and multimedia applications and the resources and services required to satisfy them. XQoS is not intended to replace current architectures, but to integrate traditional communication services (i.e. UDP and TCP transport services over Best-effort IP networks) with new specialized services (i.e. new transport and network services) sharing the XQoS semantic namespace.

Next sections are organized as follows. Section II introduces the XQoS framework. In section III, the FPTP XQoS aware transport protocol is presented. Section IV shows a study case and experimental results. Finally, the conclusions and perspectives of this work are presented.
specifications. Next section introduces the language specification proposed by the XQoS architectural framework.

B. XQoS Specification language

The XQoS specification language is intended to describe the user and applicative requirements in terms of quality of services. This language is also aimed to specify the available services and resources of the communication system. On one hand, the QoS requirements should completely describe the constraints associated to the data being exchanged by the applications. These constraints must include the user preferences, the application requirements and the characteristics of the media being exchanged and their relationships. On the other hand, the available communication services and resources should also be described in order to select and deploy the adequate QoS mechanisms intended to provide, control and manage the QoS.

Next paragraphs introduce the requirements, resources and services specifications composing the XQoS language. Examples of these specifications are included in order to illustrate the capabilities offered by this language.

1. QoS requirements

The XQoS specification of requirements is intended to describe user preferences, application needs and the intra and inter flow constraints of the media participating in the communication session. Standard languages and protocols are used by current applications in order to describe the user and applicative requirements and the media presentation and session constraints (i.e. HTML, SMIL, SDP, HTTP, RTP, etc.). However, these standards do not share a common semantic that could be used to translate or map the requirements between the different system layers. XQoS is intended to offer this common semantic namespace, in particular in the boundaries between multimedia applications and the communication subsystem. At this level, two XQoS specifications are available: session specifications intended to describe the set of flows being sent or received by applications and media-type specification used to describe specific intra-flow constraints of individual media.

a) Session specification

This XQoS specification is intended to describe the QoS requirements of the flows participating in the session established between end systems. This specification must also include specific user preferences (e.g. higher priorities for some flows, target and minimal quality requirements, etc.). An XQoS session specification must include the following QoS characteristics:

- Every XQoS session must start describing the inter-flow synchronization constraints. These constraints are defined by the elementary synchronization operators: \texttt{par} (parallel), \texttt{seq} (sequential) or \texttt{excl} (exclusive).
- The QoS requirements of each flow participating in the session are specified using specific attributes of the \texttt{flow} element (i.e. order, reliability, bandwidth, latency or jitter).
- When a flow is built up of data packets presenting different QoS constraints, the description of the characteristics of these sub-sets of flows is done using the \texttt{sub-flow} element (e.g. reliability or priorities of the sub-flows of \texttt{i}, \texttt{p} and \texttt{b} pictures of MPEG flows). The \texttt{aduMask} attribute allows the identification of the different sub-flows.
- The relation between the different media flows composing the session in terms of synchronization, user preference, and inter-dependency constraints can also be specified.

Figure 2 shows an XQoS session document describing the requirements of a multimedia presentation composed by one audio and one video flow. The audio flow asks for non elastic QoS described by a fully ordered and fully reliable service, a constant bandwidth of 32 kbps and other parameters derived from the specific media-type specification (i.e. latency, jitter). The video flow is associated to the MPEG media-type definition, composed by 3 sub-flows of \texttt{i}, \texttt{p} and \texttt{b} pictures with specific QoS characteristics. This specification describes the elastic QoS comprised between a target variable bit rate of 945 kbps (full video) and a minimum service derived from the partial reliability accepted over the \texttt{p} and \texttt{b} pictures. The GOP pattern is specified by the repeat attributes for each sub-flow. Detailed sub-flow requirements will be presented in the XQoS media-type specification.

b) Media-type specification

XQoS media-type specifications are intended to describe the structure of standard media such as images, documents, audio, video, etc. A \texttt{mediaType} element corresponds to a flow element with formal QoS parameters representing the bit stream composing the media. Formal parameters can be replaced in an XQoS session specification with actual
parameters. A mediaType can be composed by several sub-flows. Figure 3 shows a definition document for the RTP Payload Format for MPEG1/MPEG2 Video described in the IETF RFC-2250 [7].

This MPEG definition describes the GOP pattern composed by the three sub-flows of i, p and b pictures. The ADU mask is specified for each flow following the standard RFC specification. The dependency relationships between the sub-flows, several formal parameters and default values have also been specified.

2. Resources specification

An XQoS resources specification describes the available resources of the systems participating in the communication session. This specification is intended to describe the physical resources (i.e. RAM memory, display, bandwidth or CPU) as well as logical resources (i.e. codec, transport and network services) of the end system or intermediate nodes.

Figure 4 shows an example of an XQoS resource specification describing the resources of an end system represented by a personal digital assistant (PDA).

III. FPTP: FULLY PROGRAMMABLE TRANSPORT PROTOCOL

A. Definition

FPTP is a connection oriented and messages oriented transport protocol. FPTP offers a partially ordered, partially reliable, congestion controlled and timed-controlled end-to-end communication service [2]. FPTP has been designed to be statically or dynamically configured according the QoS requirements. FPTP services are implemented by the composition of configurable mechanisms suited to control and manage the QoS. The compositional architecture of FPTP and its programmable and configurable characteristics will be introduced in next paragraphs.

B. FPTP compositional architecture

Different compositional frameworks have been proposed to build up communication architectures. Most of these frameworks follow a hierarchical composition model. In these models the communication system is designed as a stack of directed graph of processing modules exchanging data or control messages. The V_STREAMS was one of the first systems supporting such composition model [12]. A stream is a full duplex connection between several linearly connected processing modules with data flowing in both directions. The basic user operations consist in write and read messages. Each processing module consists of a pair of queues, one for each direction. In the V_STREAMS system a stream may be dynamically extended by the addition of new modules. Another compositional model called the x-kernel system supports a more general directed protocol graph implemented in an object based framework [8]. X-kernel offers a flexible mechanism for configuring a protocol graph, which makes it easy to plug protocols together in different ways. This protocol graph is statically configured at initialization time. Other architecture frameworks are based on a non-hierarchical composition model, where there is not any mandatory linear order between processing modules. An example of this model is the Cactus system [5]. Cactus offers a framework to build customized protocols based on the composition of specialized micro-protocols. The configurable transport protocol (CTP) has been designed and implemented using the Cactus framework. The ADAPTIVE communication environment (ACE) introduces another non-hierarchical approach for constructing configurable protocols [13]. ACE automates communication software configuration and reconfiguration by dynamically linking services into
applications at run-time and executing these services on one or more processes or threads.

FPTP architecture follows a hierarchical model for the composition of services related to QoS control mechanisms and a non-hierarchical model for the QoS management mechanisms (see figure 5). Control operations are executed in synchronization to every data packet being sent or received. Received data packets compose the IN flow and sent data packets the OUT flow. The QoS control mechanisms operate on each individual data packet in order to satisfy the QoS requirements (i.e. rate control, flow control, time control, loss detection, etc.). Management mechanisms operate in longer periods of time reacting to QoS measures or when specific events are triggered (i.e. congestion control, error recovery, inter-flow synchronization, etc.).

C. XQoS Framework and FPTP

FPTP is XQoS-aware which means that:

- FPTP services are oriented to the user and applications requirements specified by the XQoS session specification.
- FPTP mechanisms operate over the application data units (ADUs) taking advantage of the QoS information specified by applications for every ADU and conveyed by the XQoS media-type specification.
- XQoS service and resource specifications are used to select the download, composition and deployment of the FPTP mechanisms. XQoS service specifications allow on one hand to describe the characteristics of FPTP services and on the other hand to specify the mechanisms to be composed and deployed in order to deliver such QoS oriented services.

Next paragraphs present a study case intended to illustrate how the FPTP congestion control mechanism can be deployed and configured in the framework of the XQoS architecture.

IV. STUDY CASE

A. TFRC congestion control mechanism

The TCP-friendly rate control (TFRC) is a receiver-based congestion control mechanism that provides a TCP-friendly send rate while minimizing abrupt rate changes [4]. The sender sends a stream of data packets to the receiver at some rate. The receiver sends a feedback packet to the sender roughly once every round-trip time (RTT). Based on the information contained in the feedback packets, the sender adjusts its sending rate in accordance with the TCP throughput equation to maintain TCP-friendliness [11]. If no feedback is received from the receiver in several RTTs, the sender halves its sending rate. This congestion control mechanism has been implemented as a FPTP service (see figure 7).

This service is composed by five processing modules. The RateControl and ProcessIn mechanisms have been deployed as control mechanisms in synchronization to the OUT and IN flow respectively. The RateControl mechanism limits the data packets being sent according to the allowed sending rate \( T \). The ProcessIn mechanism processes each data packet being received and sends a local management signal when a loss is detected. The CreateFeedback, ProcessFeedback and NoFeedback mechanisms have been deployed as management mechanisms. The CreateFeedback mechanism produces and sends a feedback message once by RTT, when a loss is detected (loss signal) or when one message is received if the sender is sending at a rate of less than one packet per RTT. The ProcessFeedback mechanism is triggered by a signal coming from the ProcessIn mechanism when a feedback
message is received. Feedback message is used to increase or decrease the sending rate. The Nofeedback mechanism operates when the no-feedback timeout expires resulting in halving the sending rate T.

B. FPTP-TFRC service and XQoS

TFRC congestion control mechanism is intended to prevent applications from exceeding the limits of network resources. However, the QoS constraints of data flows being controlled are not taken into account by TFRC. In order to demonstrate the advantages offered by the XQoS framework and the programmable and configurable capabilities of the FPTP protocol, the TFRC mechanism has been enhanced in order to take into account some QoS requirements of the multimedia flows.

The TFRC rate control mechanism operates by delaying the data packets according to the sending rate limit. This mechanism can penalize some applications with strict latency constraints such as interactive applications. An alternative to the delaying policy implemented by TFRC is the reduction of the required flow bandwidth. This alternative strategy can be implemented by respecting the latency time constraints. The reduction of the flow bandwidth can be mainly achieved by two mechanisms: data flow re-encoding or selective discarding. First one is considered as an application layer mechanism and it is usually not advised to be executed at the transport level. In contrast, the selective frame discarding can be implemented as a transport layer task. This mechanism must be implemented minimizing the quality degradation perceived by the user. This optimization may be performed if detailed QoS information on data packets is known (i.e. information provided by XQoS session and media-type specifications). These considerations led us to propose a new FPTP-TFRC service QoS oriented characterized by: (see figure 8):

- Enhancement of the TFRC rate control with a QoS-aware mechanism based on selective discarding (i.e. RateControl_QoS-aware).
- Incorporation a new control mechanism able to recover the applicative QoS information associated to each ADU (i.e. QoSParser). The QoS information related to each ADU should contain presentation time, segmentation information (i.e. segmentation of pictures in a video flow) and ADU types, priorities and interdependency relationships (i.e. i and p picture for H.263 flows or i, p and b pictures for MPEG flows).
- Development of a new management mechanism enabling the activation of the adequate rate control mechanism in response to the latency requirements of the application and the latency measured during data transmission.

C. Experimentation

Experiments have been performed in order to illustrate the advantages offered by the XQoS aware services offered by FPTP. For these experiments only a Best-Effort network service has been considered. Two kinds of applications presenting different QoS requirements in terms of time constraints have been studied: streaming and interactive applications. In [10] the user requirements in terms of maximum latency tolerated for the streaming applications have been shown to be around to 10 seconds. For interactive applications this parameter is considered as being around to 150 ms with a maximum of 400 ms. XQoS specifications describing the requirements of these applications and the available network services have allowed to select the adequate QoS mechanisms to be deployed by FPTP\(^1\). For streaming applications presenting flexible latency constraints the standard TFRC congestion control has been selected and deployed. For interactive applications with strict latency constraints, the TFRC QoS-aware congestion control has been selected.

![Fig 8. FPTP-TFRC QoS-aware service](image)

![Fig 9. FPTP-TFRC service for streaming applications](image)

Figure 9 shows the results of the first experiment involving streaming applications. Figure 9.a illustrates the bandwidth required in bytes per second for a RTP/MJPEG flow streamed without any congestion control mechanism. Figure 9.b shows the effects of the FPTP-TFRC congestion control

\(^{1}\) These mapping mechanisms are performed using XML facilities offered by XQoS which are out of the scope of this paper.
mechanisms on the same flow. During the first 7 seconds of the slow-start phase, the ADUs produced by the application exceed the sending rate authorized by the TFRC rate control mechanism. These ADUs are temporarily buffered and transmitted later according to the exponential increase of the sending rate. After 7th second, the allowed sending rate becomes higher than the MJPEG flow rate and next ADUs being produced by the application, will not be delayed. Therefore, the slow-start phase of the standard TFRC congestion control may be considered as being compliant with the latency constraints of streaming applications.

The FPTP-TFRC QoS-aware service to be deployed for interactive applications. Figure 10.b shows how this rate control QoS-aware mechanism discards some i and p pictures of the H.263 flow during first 3 seconds in order to respect the sending rate limit. Figure 11.b shows that during first 4 seconds only some i pictures are sent, between 5th and 6th seconds only i and p pictures are transmitted and after 6th second the full MPEG flow start being sent. These experiments demonstrate the advantages offered to legacy multimedia applications by the FPTP protocol in the framework of the overall QoS semantic namespace provided by XQoS.

VI. CONCLUSIONS AND PERSPECTIVES

In this paper the FPTP transport protocol and the common semantic namespace of the XQoS framework have been presented. This context has allowed specifying and implementing the advanced and QoS oriented services of the Fully Programmable Transport Protocol. Study cases have demonstrated the feasibility and advantages of this approach for legacy applications over a Best-Effort network service. Other QoS oriented services offered by FPTP are being evaluated and deployed in the XQoS framework. Research studies about the dynamic reconfiguration of FPTP control mechanisms and their interaction with other transport and network services are currently being carried out (i.e. multicast, mobility, FEC, caching, active networks, DiffServ).

VI. REFERENCES