QoS in real time data transmission

Hugo Pereira, Marília Curado, Paulo de Carvalho

Centre for Informatics and Systems, University of Coimbra
Department of Informatics Engineering, University of Coimbra
Pólo II, Pinhal de Marrocos, 3030 Coimbra, Portugal
Email: {hmanuel, marilia, carvalho}@dei.uc.pt

Abstract: *The increasing demand for reliable and available network services is becoming a concern. A costumer will choose an Internet Service Provider (ISP) based on the decisive criteria of the available services and enabled functionalities, i.e., based upon the QoS provided.. QoS can be regarded as the ability of an application to obtain the network service it requires for a successful operation. QoS is particularly critical for applications with real time requirements, such as applications involving tele-conference. Under these circumstances, the selection of an appropriate transport protocol exhibits a significant impact on the QoS perceived by the user. In this work, is presented an overview of some existent QoS models and of some transport layer protocols. Furthermore, performance tests over different networks using RTP are introduced. In order to define the "overhead" that RTP presents when transmitting multimedia data, comparative tests using UDP packets are also presented.*

Keywords: QoS, Transport Protocols

1. Introduction

In a network there are some quality requirements in order to successfully run an application. These requirements are commonly referred to as Quality of Service (QoS). The term QoS is a very subjective term. In fact there is not a common agreement on its definition. For a normal user, the lack of QoS may be simply when a particular remote application does not behave as expected, either by lack of correct feedback or by excessive response latency. However, for a more technical user this can be a set of mechanisms that enable a good data flow between two peers. According to the definition introduced by ITU, QoS is "*a service provided by the service plane to an end user (e.g., a host [end system] or a network element) and which utilizes the IP transfer capabilities and associated control and management functions, for delivery of the user information specified by the service level agreements*" [1]. An optimal quality of service is the one in which the behaviour of data transmission in a network (e.g. access to a file) would be the same as in a local disk: for instance, the user would read a remote file like if it was reading from a locally stored file.

QoS is the umbrella of a wide set of standards and mechanisms that ensure a good performance for network traffic quality applications. This policies or QoS rules are usually managed by network administrators in routers or proxies that filter and manage the traffic that access the network. Also all other elements through which a traffic flow passes – network interface cards, switches and bridges – must support QoS. If one of these devices along this pass does not support QoS, the traffic flow[1] will receive the standard first come, first served treatment existing in that network area.

QoS guarantees that the transactions performed by an application throughout a network, may be processed within an acceptable amount of time [2]. QoS may be used to handle UDP or TCP traffic in order to manage the priority of applications that rely on these protocols, so that the required bandwidth may be available, even during network congestion. With a traffic contract (SLA – service level agreement) it is possible to specify a mutual agreed measure of network throughput, performance and latency, accomplishing specific network application needs like multimedia streaming or more safety critical applications (e. g. remote surgery that requires a high level of availability).

Figure 1 shows how QoS is present through the TCP/IP layer model.

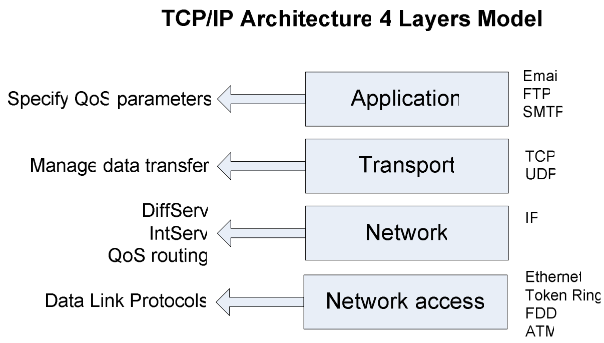

Figure 1- TCP/IP layers in QoS.

In the application layer are specified the QoS policies. Usually in a wide area network, a service provider enables several mechanisms, namely transit delay, residual error probability, priority, source routing, congestion control, sequence preservation probability

[1] A traffic flow or data flow identifies a set of packets that receive special QoS.

and maximum packet lifetime. The transport layer provides a transparent data transfer between hosts, relieving the upper layers from any concern in data transmission. In this layer there is not a major concern in QoS research. Nevertheless, there is some exploratory work in this area, such as the Application Oriented Transport Protocol (AOTP) introduced in [4] [5]. This protocol intends to provide priority based error recovery, multimedia playback management and adjustable partially reliable service. The main target of this protocol is multimedia applications with a better performance in wire line networks. Support for wireless networks is not foreseen, due to their abrupt latency variations.

In the network layer there may be applied two QoS models currently being standardized within IETF: Integrated Services (IntServ) and Differentiated Services (DiffServ). These two types of services enable a better management over the existent network resources and over the data flow, increasing the quality of service. They will be explained further in the remaining of this document. This layer is also responsible for QoS routing [6] where it is selected the network path that satisfies the required constrains: having multiple network paths with the same destination, it is chosen the one that may offer better conditions for data transmission (e. g. congestion detection, packet loss probability).

Delivering data in real time (e. g. video) over the internet is a complex issue: traffic may be processed as quickly as possible; but there are problems due to packets drop or late arrivals, giving few guarantees of delivery. Currently Internet provides best effort service and is limited by its bandwidth, delay and loss. Because of end systems heterogeneity, the difficulty to multicast or unicast real time data in an efficient and flexible way is increased. The unicast delivery of real-time data uses point-to-point transmission, with only one receiver and one sender. In multicast delivery of real-time data there is point-to-multipoint transmission, where there is one sender and several receivers.

Network administrators can use QoS to manage UDP and TCP traffic. Unlike TCP, UDP is an unreliable protocol that does not receive feedback from the network. Hence it is not able to detect network congestion. With QoS it is possible to manage the priority of applications that are based on this protocol, so that they have the required network quality like bandwidth, without increasing the congestion problem. Bypassing this kind of problems with good QoS policies, gives administrators control over the network resources, ensuring that real-time and critical applications may be successfully executed without congesting even more the network. From the financial point of view, QoS reduces costs by using resources efficiently, delaying or reducing the need for expansion or upgrades [7].

This paper presents an overview of the available network protocols that may be utilized to ensure high quality performance for critical applications, using the existing network resources. Here will be given a special emphasis to real-time protocols, namely RTP/ RTCP (Real-time Transport Protocol/ RTP Control Protocol), SCTP (Stream Control Transmission Protocol), POC (Partial Order Connection) and DDCP (Datagram Congestion Control Protocol), UDP and TCP. In section 2 is presented and described several QoS models. Section 3 presents transport layer protocols that have direct influence on QoS. In section 4 a set of tests performed during the present study in order to simulate streaming applications like NetMeeting and Cisco IPTV are introduced and discussed. Finally in section 5 it is presented the final notes and conclusions about the tests.

2. QoS Models

A QoS model comprises several mechanisms to achieve QoS, including policies, scheduling, queue management, admission control and resource reservation The following sections reference several QoS models, namely, IntServ, DiffServ and MPLS, highlighting their main advantages and disadvantages.

2.1. Best-Effort

A best effort [8] network does not deliver the performance required for a wide range of interactive and multimedia applications that have demanding delay and bandwidth requirements. Since in the best effort model there is not resource allocation, the routers and switches on the data path do not store state information concerning reservations. All requested connections are admitted and the available resources are shared among the connections. The source has the responsibility to define how much should be sent. In best effort networks there is not differentiation between traffic flows. This type of service is adequate when the load is low and when there is no particular sensitive application injecting traffic in the network. However, the behaviour of a data stream in the presence of congestion is completely unpredictable: there is no guarantee that a critical application will perform correctly. Aggressive flows (e. g. multimedia data stream like videoconference) will tend to monopolize network resources, and less aggressive flows may suffer starvation.

Because of the lack of quality of service guarantees, the traditional IP delivery model is referred as best effort, with a point-to-point protocol such as Transmission Control Protocol (TCP), providing some reliability to a connection flow by using some mechanisms, such as packet retransmission. However, these mechanisms have the drawback of introducing additional traffic in the network and of increasing delay.

An approach for the management of best-effort traffic in heterogeneous WANs (Wide Area Networks) is presented in [9]. It is proposed a traffic monitor that takes some statistical performance metrics and uses them to validate negotiated SLAs (Service Level Agreements).

2.2. Integrated Services (IntServ) and RSVP

The Integrated Service/RSVP [10] architecture is influenced by the work of Ferrari [11]. This model defines several service classes that, when supported by all routers and switches throughout where the data stream is transmitted, can support certain QoS requirements. By relying on resource reservation, the records of the allocated resources are maintained for each connection request. In this model the following service classes are defined:

- **Guaranteed Service [12]:** for applications that require rigid delay constrains, where it is important to reserve a certain level of bandwidth. It is intended for real time data flows, such as audio and video applications that use playback buffers and are intolerant to packets that arrive after their playing time. For a specific data flow, the router needs to be informed of the traffic characteristics (TSpec) of the stream, and informed with the reservation characteristics (Rspec).
- **Controlled Load Service [12]:** for applications that require reliable and enhanced best effort service. A TSpec with the traffic characteristics of the data flow must be submitted to the router as for the case of guaranteed service. After a data stream is accepted for controlled load service, the router assures a service equivalent to a best-effort on a lightly loaded network. The performance of a flow of the controlled load service does not deteriorate as the network load increases.

The philosophy of this model is that the routers are able to reserve resources in order to carry out the QoS requirements. However, in order to accomplish this, it is needed to keep the data flow state information in the routers.

This model is suitable for applications with dynamic QoS requirements (e. g. the frame rate in a videoconference session that can be dynamically altered according to the packet loss ratio) but there is however a scalability issue: single routers dealing with many simultaneous flows will become bottlenecks, leading to the degradation of QoS. When working with many simultaneous flows, it must be guaranteed enough routers to manage the processing overhead.

2.3. Differentiated Services (DiffServ)

The Differentiated Services model was introduced [11] to overcome the complexity associated with Integrated Services and RSVP. This model [13][14] does not use point-to-point communications to reserve resources as in the case of the Integrated service model. It is applied to a single point or region of the network, and uses a system called Behaviour Aggregate (BA) classification to group packets into classes based on predefined rules: The packet class is defined in a field of the packet header called DiffServ Code Point (DSCP). There are a limited number of services indicated by the DSCP field, and, therefore, the amount of state information is proportional to the number of classes rather than to the number of flows. The treatment given to a packet with a particular DSCP is called Per-Hop Behaviour (PHB) and is managed independently at each network node. Sophisticated classifications, marking policing and shaping are only needed at the boundaries of the network.

Internet Service Providers (ISPs) routers need only to implement BA packet classification to differentiate between the several data flows. In order for a costumer to receive a differentiated service, it must have a Service Level Agreement (SLA) with its ISP. The SLA specifies the supported classes and the amount of traffic allowed in each class. The SLA may be static or dynamic: static SLAs are negotiated periodically (such as monthly or yearly); the costumers that use the dynamic SLAs must use a signalling protocol to request services. In order to provide QoS, the required resources are configured in the routers based on the SLAs. Point-to-point QoS can be ensured by the concatenation of all SLAs of the neighbouring domains in the transmission path, resulting in independent PHBs concatenation. When a packet moves from one domain to another the differentiated service packet header fields may be remarked according to the SLA between the two domains. The traffic aggregation model allows for the scalability of this model.

However, besides having good scalability characteristic, DiffServ assumes a static SLA configuration between the customer and the provider. In the real world there are heterogeneous topologies that change very rapidly. Real time traffic, such as videoconference, requires per flow guaranties while DiffServ only provides guaranties for the aggregates.

2.4. Multi Protocol Label Switching (MPLS)

MPLS (Multi Protocol Label Switchig) [30] has been developed and standardized by the Internet Engineering Task Force (IETF) with the objective of reducing the complexity of IP forwarding. With MPLS routers it is not needed to perform an address lookup for every packet, speeding up the packet forwarding time.

Basically, in a MPLS network a label is assigned to incoming packets by an edge router. These labels contain information based on the routing table entry (destination, bandwidth, delay) and refer to the IP header field (source IP address), socket number information and differentiated service. Once this classification is complete, the packets are assigned to the corresponding Label Switched Path (LSP). At each hop, the Label Switched Router (LSR) applies a new label for the next hop with updated information. The Class of Service (CoS) field (EXP) is used to determine the type of treatment to be applied, like queuing and scheduling. So, different packets may receive different treatments along the path to the destiny. This approach is known as experimental bit inferred label switched paths (E-LSPs), indicating that the QoS information is inferred from the EXP field.

Another approach for QoS support in MPLS networks is the label inferred label switched paths (L-LSPs). The QoS information is obtained from the MPLS label, where all packets entering in the LSP are applied with a fixed CoS value. The label associated with a MPLS packet specifies how a packet should be treated.

With these two MPLS approaches it is provided QoS controls for service delivering, by allowing dedicated paths to be set up, and bandwidth reservation along the same path, by using explicit LSP. So it is allocated network resources to traffic according to their requirements.

MPLS can not be considered a QoS model, but a mechanism that, when working in conjunction with other QoS architectures like IntServ or DiffServ, may provide the required levels of end-to-end QoS management in a scalable way. In [17] it is proposed a scheme with a combined approach of MPLS and DiffServ.

3. Transport layer protocols

In the following section transport layer protocols for QoS are presented, namely RTP, SCTP and DCCP. The impact of these protocols on the transmission of real time traffic is specially addressed.

3.1. Real Time Transport Protocol/Real Time Control Protocol (RTP/RTCP)

The Real Time Protocol [16] (RTP) is a point-to-point protocol used to carry multimedia traffic, namely audio and video, over IP networks. This protocol provides also network transport functions intended for applications with real time requirements, videoconference or simulation data, over multicast or unicast services.

The Transmission Control Protocol (TCP) in the most widely used transport level protocol in Internet. However it is not suitable for real time applications, because it includes a retransmission mechanism which is useless for critical applications like in videoconference; it is a point-to-point protocol without direct support for multicast transmission and there is no timing information available which is required for most real time applications. The other widely applied transmission protocol, the User Datagram Protocol (UDP), also does not include timing information. RTP was specified within the Internet Engineering Task Force (IETF) to fill the gaps of UDP, namely the unreliable and connectionless transmission of data through IP networks. The International Telecommunications Union (ITU) has adopted RTP as the multimedia transport protocol. Also the ITU-T recommendation H.323 includes RTP as the transport protocol for multimedia sessions.

Packets sent on the Internet have an unpredictable delay and jitter. Multimedia applications require appropriate timing in data transmission and playback. RTP provides time stamping, sequence numbering and other mechanisms to take care of timing issues. Real time data transmission is assured through these mechanisms.

The main RTP characteristics are the following:

- RTP provides end-to-end delivery services for data with real time characteristics, such as interactive audio and video. However, RTP does not provide any mechanism to ensure timely delivery: it needs support from lower layers that have direct control over resources in routers and switches namely RSVP to provide the required resources.
- RTP does not assume anything about the network layer, except that it provides framing. Typically RTP runs on top of UDP, making use of its multiplexing and checksum service. However efforts have been made to make RTP compatible with other network protocols like ATM, AAL5 and IPv6.
- RTP does not offer any reliability or flow congestion control. It provides tools like time stamps and sequence numbers to the application layer in order to implement mechanisms for reliability.
- RTP is a modular protocol: by adding a new profile and a payload format it may be integrated into new formats and applications.

The RTP data transport is improved with a control protocol (RTCP), which provides feedback on the quality of the data transmission to the RTP session. The transmitted packets must be multiplexed into data and control packets. With UDP this is normally implemented using separate port numbers.

The RTCP packets contain information about QoS monitoring and congestion control, session size estimation and scaling. The sender and the receiver exchange information about packet losses, delay and jitter. Using a network management tool it is possible to know the actual network state without receiving data packets, based only on RTCP packets. The more participants there are, the more RTCP packets are exchanged. Therefore, this kind of traffic control must be limited: there must exist a trade-off between the amount of real time data and traffic control data. Usually, it is scaled for about 5% of the total generated traffic.

Basically, the RTCP appears as a solution that gives reliability to a RTP data flow. RTP/RTCP provides functionality and control mechanisms necessary to carry real-time content, done at the application level. The flow control congestion information is provided by the RTCP sender and receiver reports.

Currently there are some RTP/RTCP open implementations, namely the RTP library API "RTPLib" [17] of Lucent Technologies, or a more recent of Vovida Software the "rtp-1.5.0" [18].

3.2. Stream Control Transmission Protocol (SCTP)

As a transport protocol, SCTP [19] is the equivalent to TCP or UDP. It provides some similar services as TCP, ensuring reliability, sequential transmission of messages with congestion control. While TCP is byte oriented and UDP is connectionless, SCTP is connection oriented.

SCTP is a reliable transport protocol operating on top of an unreliable connectionless packet service such as IP. It provides acknowledgements, error free, non duplicated transfer of messages throught the use of checksums, sequence numbers and selective retransmission mechanism. SCTP has adopted flow control and congestion from TCP and presents new features that make it more suitable for signalling purposes than TCP. The most interesting characteristics of SCTP is multi-homing support and the ability to use several separated streams inside an association/connection. Each stream can be delivered in an ordered or unordered way if the user application wishes. The support for multi-homed nodes in SCTP implies that the remote peer can be reached using more than one IP address. If it is guaranteed that, for each IP address used for the same destination, the data stream travels through different physical paths, the association becomes tolerant against physical network failures. The information about the multiple addresses is exchanged at the time of the association setup. One of the addresses is selected as the primary path over which the datagrams are transmitted by default. Retransmissions can be done using one of the available paths.

Instead of the three phase connection setup of TCP, the initialization of an association is completed after the exchange of four messages. The passive side of the association only allocates resources for the association for data transmission until the third of these messages arrives and has been validated. This exchange of packets is a feature of SCTP that gives more resistance to issues of DoS (Denial of Service) attacks, by using encryption algorithms like MD5.

After the association is established, each transmitted data chunk is numbered with a Transport Sequence Number (TSN) to enable detection of loss and duplication of data packets. A Stream Sequence Number (SSN) is assigned to each datagram in order to alow for the reliable delivery of datagrams.

Heartbeat packets are sent periodically to track the availability of the network. These packets are used to check out the available connections, and each heartbeat is acknowledged by a heartbeat ACK always through the default address. If a transmission over a certain path fails repeatedly, the path is regarded as inactive. However, heartbeats still continue to be sent to the inactive addresses. There must exist a trade-off between the heartbeat packets, that give information about the network state, and the data flow transmitted: heartbeat traffic can be dynamically configured according to the needs of the application. The application can disable and re-enable heartbeat, change time interval and request for an "on demand" heartbeat.

Currently, an open source library is available that implements this protocol: *sctplib* [20].

3.3. Datagram Congestion Control Protocol (DCCP)

DCCP [21][22] is a message oriented transport layer protocol currently under development in the IETF. It aims to be a substitute of UDP, and to be used as a real time transport protocol, with timing constrains for data delivery. The primary motivation for the development of DCCP is to provide a way for real time applications to gain access to standard congestion control mechanisms without having to implement them in the application layer.

For real time data transport, UDP is more advisable mostly because of its fast delivery. However, it exhibits problems of congestion, which translates on packet dropping and out of order packet reception. From the application point of view, firewalls and NAT's (Network Address Translation) do not always pass UDP traffic. The solution is to implement UDP with congestion control and handshake during connection setup and termination. This is where DCCP enters. It provides the following features:

- A reliable negotiation of features.
- A choice of TCP friendly congestion control mechanisms, including TCP like congestion control and friendly rate mechanisms. It is appropriate for data flows that want to quickly take advantage of the available bandwidth, and can change send rates dynamically.
- Congestion control incorporates Explicit Congestion Notification (ECN). During a DCCP connection establishment, acknowledgment packets are sent to inform whether the packets have arrived and if they are ECN marked. This marking gives reliability to the data transmission flow.
- It has options that tell the sender, with high reliability, which packets reached the receiver and whether those packets were ECN marked, corrupted or dropped in the receive buffer.
- Incorporates mechanisms allowing a server to avoid holding any state for unacknowledged connection attempts or already finished connections

At the beginning of a DCCP connection, the end points must agree on a set of parameters, namely the congestion control mechanisms to be used. DCCP provides a minimal set of options for negotiating the values of the general features, where a feature is simply a value meant to be negotiated.

One of the features of DCCP is the possibility of choice between TCP friendly congestion control mechanisms (CCID 2) and TCP friendly rate control (CCID 3) [23]. CCID 2 is appropriate for flows with abrupt frame rates: it uses a sender congestion window to limit the number of received acknowledged packets, apply timeouts and responds quickly to changes in the available bandwidth. However, it inherits the TCP slow start that is required for establishing a reliable connection. CCID 3 is appropriate for flows that require a steadier sending rate: the receiver sends to the sender feedback, once per round trip time, with the loss event rate calculated at the receiver. This feedback is used at the sender to calculate a new sending rate. If no feedback is received from several round trip times, the sender cuts to half its sending rate. This tries to minimize abrupt changes in the sending rate.

This communication protocol is still very recent. The first IETF draft publishing of DCCP was in 2002. However, there are already two alpha implementations available of this protocol, one described in [24] and the other presented in [25].

4. RTP experimental tests and results

The tests presented in this section were performed using the RTP protocol, because it is widely spread and it is supported by a wide range of applications. The behaviour of this protocol is assessed controlled network conditions and in non controlled networks, namely, in a 10Mbps LAN, using an ADSL connections (128Kbps upload, 2Mbps download), and a cable (128Kbps upload, 256 download) internet connection.

IxChariot [26] from Ixia was applied to perform these tests and to measure the performance between pairs of networked computers under distinct networks. This application retrieves several network performance parameters like jitter, losses and delay. Moreover, it allows for the simulation of certain multimedia applications, namely Cisco IP/TV and NetMeeting audio and video.

4.1. NetIQ Chariot

Chariot is an application that measures network performance, by emulating data transmission in a network using several protocols, including RTP. IxChariot™ provides the ability to confidently predict the expected performance characteristics of any application running on wired and wireless networks [26]. To perform this operation, scripts are utilized in order to mimic certain system behaviours (e. g. NetMeeting). As depicted in Figure 2, this application is composed by endpoints and a console. In each peer used in the network test, it must be installed the endpoint software, which is free of charge from http://www.ixiacom.com/products/.

In order to begin a network test, (**1**) the console sends setup information to endpoint 1, like address of endpoint 2, protocol used to connect to endpoint 2, duration of the test and how to report results. (**2**) Endpoint 1 keeps half of the application script and sends the other half to endpoint 2. When endpoint 1 has acknowledged that the other endpoint is ready, it replies to the console. Then the console recognises that the endpoint pair is ready to begin the tests, and sends a message to begin. (**3**) The endpoints begin the tests by starting their application script. At this stage endpoint 1 starts collecting the records that contain the data. (**4**) Finally, the endpoint returns the results to the console, which interprets analyses and shows them.

In the console, the obtained measurements are divided into four sections: throughput, transaction rate, response time and lost data. Throughput is the traffic generated during the test between the two endpoints. It is calculated with the following equation when using streaming scripts:

Throughput=(bytes received by Endpoint2) / Measured time

Transaction rate is defined as the ratio of each operation that involves sending a packet and receiving its confirmation (in a streaming protocol this definition is not correct, because there is no confirmation of delivery; in these cases a transaction will be assumed when endpoint 2 receives a packet). This is calculated with the following equation:

Transaction rate = Transaction Count / Measured Time

Response time is the inverse of the transaction rate:

Response time = Measured Time / Transaction Count

Lost data is the difference between the number of bytes sent in endpoint 1 and the number of bytes received in endpoint 2. This measurement is done only when running streaming scripts.

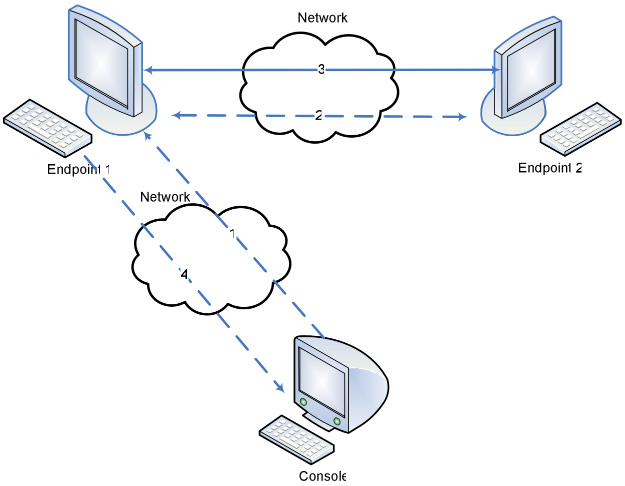

Figure 2 - Chariot data flow [26].

4.2. Test definition

Chariot emulates applications with specific scripts that are negotiated at the beginning of a test between the endpoints. In these tests Cisco IP/TV and NetMeeting audio and video applications were used.

IPTV [27] is the Internet Protocol TV that delivers television programming using the internet protocol over computer networks. Cisco developed its IPTV application for audio/video broadcasting using IP multicast, video-on-demand using RTSP and unicast RTP over IP networks [28]. Cisco IP/TV solution is based on an efficient network multicast technology with a quality streaming solution, offering control over bandwidth and network performance.

NetMeeting [29] is a conference application developed by Microsoft. It allows users to interact in real time over the Internet. Due to NetMeeting's low bandwidth requirements, it is possible to run the application using 56Kbps modem connections. However, for better performance, a faster network connection is advised.

IPTV and NetMeeting are used for different purposes, hence exhibit different QoS requirements. IPTV uses a larger bandwidth because of its high video on demand definition. NetMeeting is a less demanding application, used for videoconference purposes.

The main parameters used to characterize the behaviour of the tests performed in the network are:

- Jitter: variation in delay;
- Lost Data: percentage the number of bytes lost from endpoint 1 and endpoint2;
- One way delay: delay between endpoint 1 and endpoint 2 in a single direction, including delay factors as the codec used, jitter buffers and fixed delays. In order to synchronize the clock values in each endpoint, endpoint 2 tries to get enough clock samples from endpoint 1 to determine the round trip time;
- Throughput: measures the speed of data transfer between the two endpoints.

These tests were made by simulating the traffic generated by IPTV and NetMeeting applications. The audio and video were tested apart because the scale ranges are very different, and, therefore, a simultaneous graphical representation is not very conclusive. The tests can be divided into two parts: individual stream analysis (where IPTV audio, IPTV video, NetMeeting audio and NetMeeting video are tested individually), and multiple stream analysis (where for video in IPTV and NetMeeting are tested simultaneous streams with different rate and packet size values).

The tests were performed in order to compare traffic performance between a controlled environment – a local area network – and a not so predictable network like the internet. To accomplish the tests over the Internet, two endpoints were specified: endpoint 1 had an ADSL Internet connection with 2Mbps of download traffic and 128 Kbps of upload traffic; endpoint 2 had a NetCabo internet connection with 256 Kbps of download and 128 Kbps of upload. The tests over a LAN were possible by using the 10Mbps local area network of the Informatics Engineering Department of the Coimbra University.

4.3. Results

In Table 1 and in Table 2 the average delay jitter, lost data and number of transactions per second is shown for both test conditions. In some cases it was not possible to obtain the delay values, because the endpoint clocks could not be synchronized. In Figure 4 and in Figure 5 are presented a graphical comparison of the jitter and lost data of IPTV and NetMeeting applications in a LAN connection and in an Internet connection.

In the LAN network tests, there was almost no lost data: all data reached endpoint 2. The delay was very small, being defined as a constant value during data streaming time. This value was due to jitter buffers, fixed delays and codec encoding and decoding. The jitter had some peaks but they are not significant.

	10 Mbps LAN connection					
Tests	**Time (sec)**	**Nr trans**	**Trans /sec**	**Avg. delay (ms)**	**Avg. jitter (ms)**	**Lost data (%)**
IPTV audio	219.95	100	0.45	n/a	0.22	0.0
IPTV video	201.38	100	0.5	4.0	0.25	0.02
IPTV video [increase rate]	135.34	251	1.85	4.0	0.83	0.02
IPTV video [increase packet size]	201.25	298	1.48	5.0	1.37	0.0
NetMeeting audio	117.4	100	0.85	1.0	0.11	0.0
NetMeeting video	130.64	50	0.38	1.0	0.2	0.0
NetMeeting video [increase rate]	85.62	124	1.45	1.0	0.09	0.0
NetMeeting video [increase packet size]	201.25	298	1.48	5.0	1.37	0.0

Table 1- Comparison of the transactions made during the performed tests in a 10 Mbps LAN, over RTP

Of course these obtained values are optimal. They serve as a base reference to the following tests in non controlled environments.

In the Internet tests, the results are limited by endpoint 1 upload bandwidth (128Kbps). When this limit is reached the delay, jitter and lost data increases.

In IPTV applications the lost data and jitter are 5 times greater than in a 10 Mbps LAN. When several streams are concurrently sending packets with packets with different size, the percentage of lost data is low. However the jitter abruptly increases, justifying the low transactions realized per second. In the case of the concurrent streams with different rates the situation is similar.

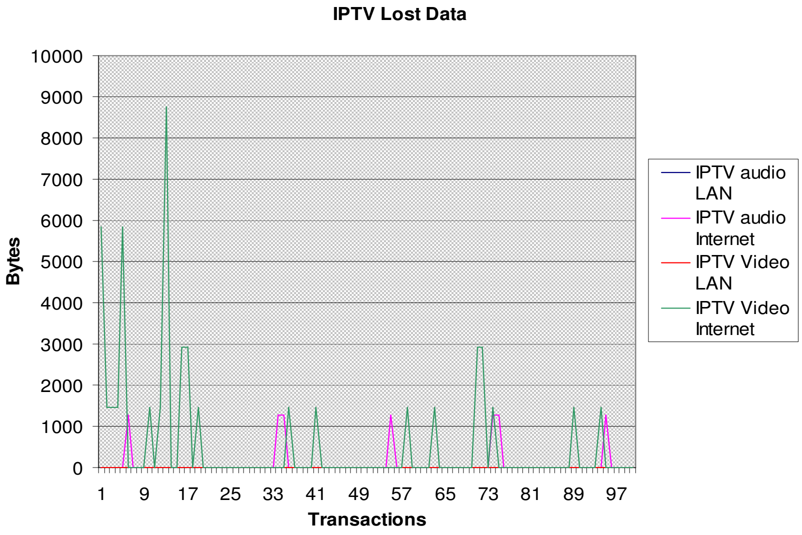

Figure 3 - Representation of the lost data by simulating IPTV application over UDP.

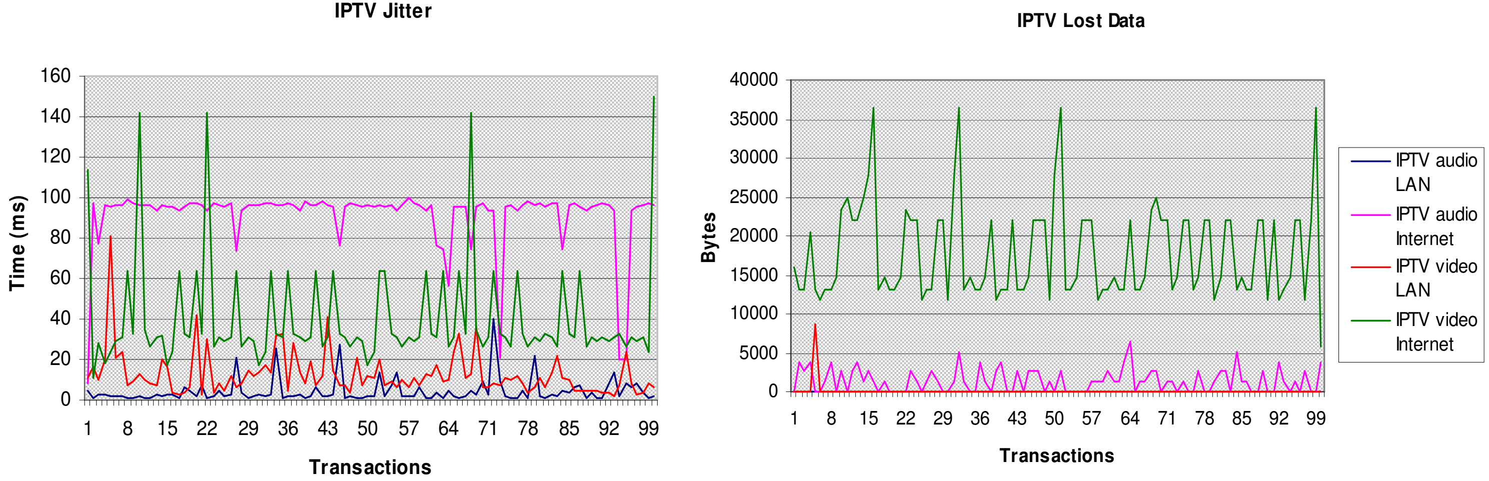

Figure 4 - Graphical results by simulating IPTV applications over RTP. In the left, it is presented the jitter (variation of the delay) and in the right it is presented the lost data.

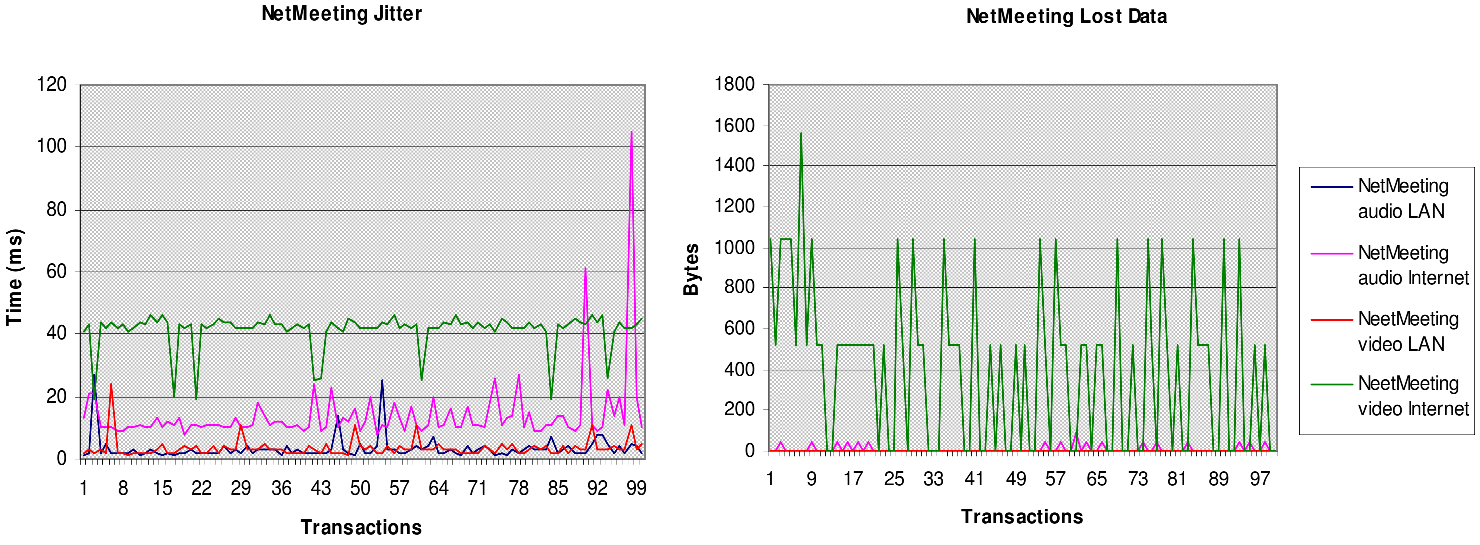

Figure 5 - Graphical results by simulating NetMeeting applications over RTP. In the left, it is presented the jitter (variation of the delay) and in the right it is presented the lost data.

	Internet connection					
Tests	Time (sec)	Nr trans	Trans /sec	Avg. delay (ms)	Avg. jitter (ms)	Lost data (%)
IPTV audio	220.02	95	0.43	n/a	6.9	5.05
IPTV video	483.42	17	0.03	n/a	6.4	5.08
IPTV video [increase rate]	2006.02	30	0.01	2.22	78.2	0.013
IPTV video [increase packet size]	3026.16	108	0.04	3.94	102.0	0.026
NetMeeting audio	117.41	100	0.85	n/a	1.6	0.45
NetMeeting video	130.65	50	0.38	n/a	2.6	2.15
NetMeeting video [increase rate]	257.78	147	0.57	n/a	6.9	1.35
NetMeeting video [increase packet size]	184.77	109	0.59	3.94	129	0.026

Table 2 - Comparison of the transactions made during the performed tests in a broadband Internet connection, over RTP

In the case of NetMeeting, when testing the audio streams, the behaviour is very similar of the one in the 10 Mbps LAN test. This happens because the 128Kbps upload limit of endpoint 1 has not been reached.

	10 Mbps LAN connection (UDP)			Internet connection (UDP)		
Tests	Time (sec)	Nr trans	Lost data (%)	Time (sec)	Nr trans	Lost data (%)
IPTV audio	219.98	100	0.0	220.01	100	0.2
IPTV video	201.40	100	0.0	201.38	100	0.24
NetMeeting audio	117.40	100	0.0	117.4	100	0.0
NetMeeting video	130.64	50	0.0	130.65	50	0.0

Table 3 - Comparison of the performed transactions tests in a 10 Mbps LAN and in a broadband Internet connection, over UDP.

When NetMeeting deals with video, the number of transactions per second decreases, and the percentage of lost data increases. When dealing with several streams, the behaviour is similar of the one with IPTV: if the concurrent streams have different packet sizes, the jitter abruptly increases (129ms) but maintaining a low percentage of data loss. If the streams have different sending rate the jitter had an inferior average value than the previous one (6.9ms), while the data loss percentage increased (1.35%). When increasing the packet size, because of the size of the packet, they reach to endpoint 2 with a big delay. The packet delivery is done, but with increasing delay costs. When increasing the rate, the number of packets to be transmitted increases. This raise lost data ratio, maintaining a low jitter.

Table 3 shows the results concerning broadband and LAN networks, by simulating the NetMeeting and the IPTV UDP packet transmission. The objective of these tests is to assess the overhead introduced by RTP.

The jitter and the delay could not be measured because the UDP packets have no time stamping fields. However the lost data percentage could be measured. In LAN networks there was no lost data, and in the broadband Internet connection there was only lost data in IPTV applications. When compared with RTP lost data average, UDP lost data is almost five times less. Therefore, the results show that the overhead imposed by RTP in data transmission is significant.

By comparing Table 1, Table 2 and Table 3, the differences are quite evident: looking into the Internet connection, the ratio in lost data is 5 times inferior in UDP streaming than in RTP streaming. This happens mainly because of the header fields that RTP have, combined with the ones of UDP. So, the packet overhead of RTP is greater than the one of a single UDP packet. Further conclusions are presented in the next section.

5. Conclusions

This document presents several models of service: best effort, integrated services and differentiated services. A best effort network is the current default model used. The DiffServ model associates several data flows into classes. It applies efficient QoS mechanisms during resource allocation. It is supported by many ISPs to set several degrees of guarantee of service: it is given a priority level to the data flows in order to increase QoS to a particular costumer. IntServ stores network state information in each network element, used to provide the requested quality of service to each flow.

Since the best effort network schema is the currently most used model, a real time application must rely on other mechanisms to overcome network flaws, namely packet drops and transmission errors. To overcome this problem there are several types of transport layer protocols. In this paper the following protocols were described: RTP/RTCP, SCTP, and DCCP.

RTP is a point-to-point protocol used to carry multimedia application traffic. However, this protocol does not give delivery guaranties. To overcome this issue it periodically sends RTCP packets in order to get information about the network state.

Because of the widely use of RTP in real time applications, this document presented performance tests obtained by simulating streaming applications based on RTP, namely NetMeeting and Cisco IPTV. The tests were done between two endpoints connected to Internet and inside a local area network.

The results showed that IPTV is not advisable for current internet connections. When the upload limit is reached, the delay starts to increase. However the percentages of lost data in those cases do not reach high values: the maximum value reached was 5.08%. This means that the UDP packets are correctly reaching their destination.

NetMeeting when compared with applications like IPTV is a good solution for real time data transmission. NetMeeting maintains a constant number of transactions per second, even when multiple streams are active. However, in these cases, the delay starts to increase.

RTP transmission, when compared to UDP transmission, has a significant overhead (about five times greater). This happens because RTP is a protocol for real-time transmission of audio and video over UDP and IP multicast: the transmitted packets have, besides the UDP headers, the RTP headers in order to enable real time data transmission.

In both studied applications, IPTV and NetMeeting, the RTP packets were received with a small percentage of loss (but with some delay). Counting with some delay spent in the transmission of data packets through the internet, RTP seems to be a wise solution for real time applications. In these tests it was only possible to test the behaviour of RTP transmission over a best-effort model provided by Internet. In order to use a better QoS model (like IntServ or DiffServ) it is needed to arrange an agreement with the ISP in order to be assigned a better level of QoS.

6. References

[1] King-Shan L.; Monet Research group; *QoS Routing;* http://cairo.cs.uiuc.edu/qosrouting/qos-routing.html; University of Illinois at Urbana-Champaign; 2000

[2] Microsoft TeckNet: *What Is QoS?* http://www.microsoft.com/technet/prodtechnol/windowsserver2003/library/TechRef/1c1f53a6-da9e-496f-be84-b91e2763dbeb.mspx; Microsoft Corporation; 2005.

[3] Zhang, J.; Stahl, J; Huang, H.; Zhou, X.; Lou, S.; Song, K.; *Real-Time Teleconsultation with High-Resolution and Large-Volume Medical Images for Collaborative Healthcare*; IEEE Transaction on information Technology in biomedicine, vol. 4, No.2, June 2000.

[4] V. Tsaoussidis, S. Wei; *QoS Management at the Transport Layer*; ITCC 2000, Las Vegas, Nevada

[5] Madisetti, V; Argyriou, A*; A new QoS Specification Is Needed For Reliable Wireless Multimedia Services*; http://www.iapplianceweb.com/story/oeg20021206s0042.htm; 2002

[6] King-Shan L.; Monet Research group; *QoS Routing;* http://cairo.cs.uiuc.edu/qosrouting/qos-routing.html; University of Illinois at Urbana-Champaign; 2000

[7] WU, D; HOU, Y.T.; ZHANG, Y.; *Transporting Real-Time Video over the Internet: Challenges and Approaches*; proceedings of the IEEE, vol. 88, no 12; December 2000

[8] Wydrowski, B.; Zukerman, M.; *QoS in Best-Effort Networks; The University of Melbourne;* IEEE Communications Magazine; December 2002

[9] Martin, J.; *Managing Best Effort IP Networks over Heterogeneous WANs*; Clemson University; 2004

[10] White, P.; *RSVP and Integrated Services in the Internet: a Tutorial*; IEEE Communication Magazine; University College London; May 1997

[11] FERRARI, D.; VERMA, D.; *A Scheme for Real-Time Channel Establishment in Wide-Area Networks;* IEEE Journal on selected areas in communications, vol. X no. 3; April 1990

[12] Laukkanen, J; *Integrated Services*; University of Helsinki; October 2000

[13] Xiao, X.; Lionel M.; *Internet QoS: A Big Picture;* Michigan State University; 2001

[14] Hao, F.;Zegura, E.; Ammar, M.; QoS Routing for Anycast Communications: Motivation and an Architecture for DiffServ Networks; IEEE Communications Magazine; June 20

[15] Sawant, A.; Qaddour, J.;*MPLS DiffServ: A Combined Approach*; Illinois state University; 2003

[16] Koistinen, T.; *Protocol overview: RTP and RTCP;* Nokia Telecommunications; 2000

[17] Lucent Technologies; http://www-out.bell-labs.com/project/RTPlib/

[18] Vovida; http://www.vovida.org/

[19] Häkkinen; A.; *SCTP - Stream Control Transmission Protocol;* Seminar on Transport of Multimedia Streams in Wireless Internet University of Helsinki, Department of Computer Science; 2003

[20] sctplib; http://www.sctp.de/sctp-download.html

[21] Kohler, E; Handley, M; Floyd, S*; draft-ietf-dccp-spec-03.txt*; May 2003

[22] xia, W.; *The Datagram Congestion Control Protocol*; For seminar Kommunikation und Multimedia in Institute of Betriebsystem und Rechnenverbunden in TU Braunschweig; 2004

[23] Kohler, E; Handley, M; Floyd, S*; Datagram Congestion Control Protocol (DCCP) Overview;* July, 2003

[24] Zawadzk, M.; Abbott, J; http://sourceforge.net/projects/dccp/; 2005

[25] MATTSSON, N.; http://www.ns.dccp.org/code.htm; Luleå University of Technology, Sweden; 2004

[26] IXCHARIOT;http://www.ixiacom.com/products/performance_applications/pa_display.php?skey=pa_ixchariot&section=perftest; 2005

[27] PRIMEDIA BUSINESS MAGAZINES AND MEDIA; What *is... IPTV?* http://assetmanagement.broadcastengineering.com/ar/broadcasting_iptv/; 2003

[28] SCHULZRINNE, H; *Cisco IP/TV;* http://www.cs.columbia.edu/~hgs/rtp/iptv.html; 1998

[29] MICROSOFT CORPORATION; http://www.microsoft.com/windows/netmeeting/; 2004

[30] NISHIMURA, K.; AIBARA, R.; BAYLE, T.; *Performance Measurements of MPLS Traffic Engineering and QoS*; Hiroshima University; Japan; 2002