Network Quality of Service Monitoring for IP Telephony

B.V.Ghita[1], S.M.Furnell[1], B.M.Lines[1], D.Le-Foll[2], E.C.Ifeachor[3]

[1] Network Research Group, School of Electronic, Communication & Electrical Engineering, University of Plymouth, Plymouth, United Kingdom
[2] Wavetek Wandel Goltermann, Plymouth, United Kingdom
[3] SMART Systems Research Group, School of Electronic, Communication & Electrical Engineering, University of Plymouth, Plymouth, United Kingdom

Abstract

This paper presents a non-intrusive method of determining network performance parameters for voice packet flows within a VoIP (Voice over IP, or Internet Telephony) call. An advantage of the method is that it allows not only end-to-end performance monitoring of flows, but also makes it possible to inspect the transport parameters a specific network or link when delay sensitive traffic transits through it. The results of a preliminary test, to check the validity of the method, are also included.

Keywords

Voice over IP, Quality of Service parameters, non-intrusive monitoring.

Introduction

Over the last two decades, the Internet has evolved from a few interconnected networks that linked research laboratories, universities, or military infrastructure, to an everyday tool which is easy to access and use by many people. The dramatic evolution can be assessed in terms of growth in the number of hosts and Internet applications. The initial use of the Internet was different to that of today. Contrasting two studies of Internet activity, from 1991 (Caceres et al, 1991) and 1997 (Thompson et al, 1997), it can be seen that the nature of activity has changed from applications such as telnet or file transfer to become dominated by web browsing (75%). The increased computational power of end-user stations has allowed new types of applications to be implemented. In addition, the speed and reliability of the Internet itself has been substantially enhanced due to the new technologies used. These advances have allowed application content to move from text to multimedia and real-time.

A major challenge in Internet development is how to support real-time applications, typified by Internet Telephony, within the existing structure. Internet Telephony aims to replace the traditional concept in telecommunications from data over voice to voice over data. The method for achieving this is to use the Internet as a transport carrier for voice, instead of the PSTN (Public Switched Telephone Network). The most obvious advantage is the low cost for long-distance phone calls.

An important barrier in the development of VoIP is the Internet Protocol (IP). IP works as a best-effort connectionless protocol. It was designed for data files that can tolerate delays, dropped packets and retransmissions; there are no guarantees about the delivery time or the reliability of a packet being transferred over the Internet. The most important aspects, when considering an audio conference are exactly those that Internet cannot guarantee: time and bandwidth. The quality of the resulting conference depends upon the satisfaction of these requirements. Within this context, the concept of Quality of Service (QoS) was introduced. Although the Internet represents an environment in which the QoS cannot be guaranteed, there are measurable parameters for a specific service, as presented in a QoS overview study (Stiller, 1995).

This paper presents an offline method of determining network performance parameters for voice packet flows within a VoIP call. An advantage of the method is that it allows not only end-to-end performance monitoring of the flows, but also makes it possible to inspect the behaviour of the network when faced with delay sensitive traffic.

QoS concept for VoIP and current state of monitoring

The QoS is the overall rating for a service. Measurement of QoS essentially includes measuring a number of application dependent parameters and then gathering them in a weighted sum. If we consider QoS for VoIP, the object of the analysis is the voice at the receiving end, with its two main characteristics, sound and interactivity. There are two main sources of impairments for the voice heard by the receiver. The first is the codec, which compresses the speech flow in order to send it over the network at a lower bandwidth than original. Aside from the positive result in terms of bandwidth utilisation, this process degrades the quality of the speech. The second source of impairment is the transport. After encoding, the audio flow is packetised and sent over the Internet. However, because of the Internet's structure, the arrival of the packets at destination cannot be guaranteed. The paper is focused upon a consideration of this latter impairment.

Building a list of performance parameters for a service should start by identifying the application that requires that specific service. For example, if the targeted application is a file transfer then the delay or jitter parameters are almost irrelevant when compared to throughput or packet loss. In a similar manner, for a real-time application, delay is far more important than the other parameters. The paper does not intend to prescribe a specific weighting here, but it is good to bear in mind their priorities when assessing the overall performance.

When considering QoS for VoIP applications, a network-related view of the performance should include the following parameters:

- delay - the time elapsed between the sending of a packet and its arrival at the destination;
- jitter - the variance of the delay value;
- packet loss - the number of lost packets, reported in the time elapsed;
- throughput - the amount of data transferred from one place to another or processed in a specified amount of time.

There are several suggested methods that can improve or guarantee the QoS for transport, such as DiffServ (Differentiated Services) (Nichols et al, 1998), Tenet (Ferrari et al, 1994), or QoS Routing combined with RSVP (Reservation Protocol) (Crawley et al, 1998). Unfortunately, none of them are applied on global basis because of the scale and complexity of the Internet. Therefore, it is vital to determine in such an environment whether or not a specific connection meets the requirements of a VoIP call.

Transport QoS has two main areas: end-to-end measurements and, in case there are changes in the level of parameters, fault localisation. An example is given in Figure 1 which shows, for an arbitrary division of the entire route of the packets, the end-to-end parameters, and two sets of parameters, 'East' and 'West'. The latter can be used to localise a fault in either 'East' or 'West' sub-network, by comparison with the end-to-end parameters.

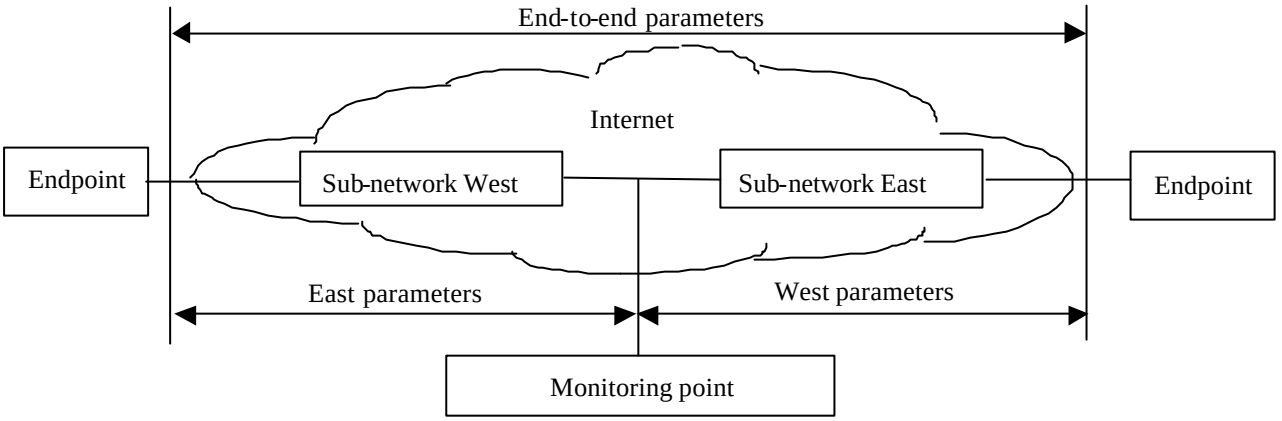


Figure 1: The Performance parameters for a general example of monitoring

In a traditional approach, the two aims would require a 3-tool configuration. For end-to-end measurements, testing clients should be put at both ends and, for fault location, a testing server should be placed at the monitoring point. After that, traffic should be collected by the end stations, then sent to the server, in order to be analysed and compared with the data collected by it. There are two main disadvantages with this approach:

- it is intrusive; in the best case, even if the endpoint clients are just monitoring, they have to send the data to the server in order to be analysed;
- it requires placement of monitoring devices at both ends.

The QoS for transport can be determined from the audio flows within a call (which run on RTP, Real Time Protocol). Current tools (e.g. Hammer VoIP Analysis System, HP Internet Advisor, rtpmon (Bacher and Swan, 1996)) base their calculations upon parsing both the RTP and/or the accompanying control flows (running on RTCP, Real Time Control Protocol) and displaying the available data. The main disadvantage is that none of these tools can establish fault location without using the traditional approach mentioned above. More than that, they do not build any relation between the end-to-end parameters, obtained from the RTCP flows, and the end-to-monitoring-point parameters, obtained from the RTP monitoring.

Considering these limitations, we aim to obtain a better view of the network performance, without using several devices and without injecting additional traffic into the network. This paper presents a non-intrusive method of determining the transport performance parameters for the real-time traffic within a VoIP call, using a single point of monitoring. The proposed method can reveal both the end-to-end performance and the fault localisation, if the monitored parameters change their value along the route, and also avoids both of the disadvantages identified.

Description of H.323 calls

VoIP is a relatively new concept and, therefore, most of the work performed in this area is still at a developmental stage. From the large range of standards for VoIP, the H.323 protocol stack (ITU, 1998), developed by ITU, was selected as the basis for the work presented in this paper.

The focus of the QoS for transport is, as mentioned, on the audio flows. Because of the H.323 call structure, which will be detailed below, these flows cannot be identified unless the entire call is monitored. The information exchanged in a H.323 conference is classified in streams, as follows: audio (coded speech), video (coded motion video), data (computer files), communication control (control data), and call control (signalling data).

We will consider the simplest case - a direct connection between two computer terminals, similar to a classic phone call. The call begins with a call signalling phase – signalling messages (Q.931 using H.225 specification) are exchanged, on specific ports. At the end of this phase, the call is established and a call control channel is opened, on ports dynamically allocated. The control channel then provides for various functions: capabilities exchange, logical channel signalling, mode preferences, master – slave determination. After the terminals decide which of them will act as a master for the call (in order to easily resolve conflicts), they exchange their capabilities and open an audio channel, using logical channel signalling. The logical channel is also opened on a dynamically allocated port, decided within the control messages. The audio flows run on the opened logical channel. When one of the users wants to terminate the call, the logical channel is closed, using call control, then specific call signalling messages are exchanged, and the call is closed.

The audio (as well as video) flows within an H.323 conference are transported using RTP, as it provides end-to-end network transport functions suitable for applications transmitting real-time data over multicast or unicast network services (Schulzrinne et al, 1996). It does not address resource reservation and does not guarantee quality-of-service for real-time services. In fact, the whole protocol is conceived not as a separate layer, but as a framework, to be integrated within other applications. RTP is usually run on top of UDP (User Datagram Protocol), an unreliable transport protocol. TCP (Transport Control Protocol), although reliable, brings additional delay problems, by delivering the packets in order and recovering the lost packets, and, therefore, is not recommended for carrying real-time flows.

RTCP is the control protocol for RTP. One of its functions is to provide information about the packets loss and inter-arrival jitter for the accompanying RTP flow. The information is provided periodically by all the senders / receivers within a conference using specific packets, and is based on the RTP flow measurements. The RTCP flow also runs on UDP.

Experimental method and implementation

Monitoring procedure

The monitoring procedure comprises three steps. First, the voice flows (RTP) are identified and then captured using one of the capture programs. In the monitoring phase, the RTP header fields and the RTCP packets are used to determine the performance parameters. Then correlation of RTP and RTCP is used to establish the location of the problem area. The stages are described in more detail in the following paragraphs.

Identification of the audio flows

The analysis is targeted on the audio streams. The ports on which the audio streams run can be determined only by capturing the connection establishment phase, then parsing the setup and control messages, which contain the audio stream ports as parameters. The parsing process is not straightforward, as the content of the setup and control messages is not header-like (using fields), but encoded using ASN.1 syntax.

Parameter measurement using RTP monitoring and RTCP parsing

The header fields of RTP packets are used as input to the analysis, together with the timestamp of the packet arrival, given by the capture program. The structure of the RTP header, as described in (Schulzrinne et al, 1996), is shown in figure 2.

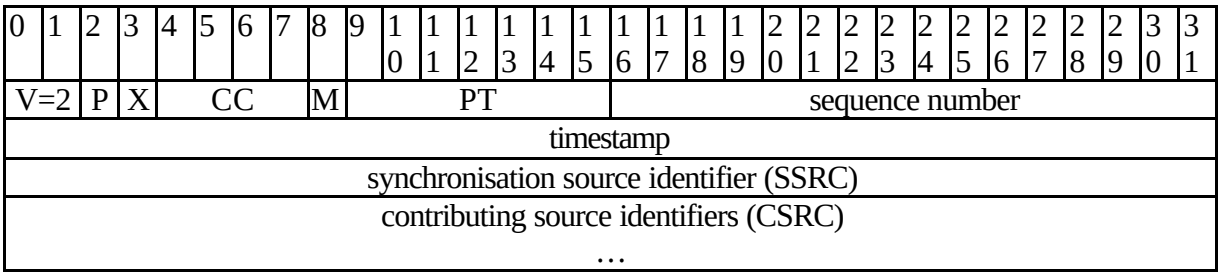


Figure 2 The RTP packet format

The description of the fields is as follows:

- V – version of RTP (currently used is 2)
- P – padding, for indicating the existence of padding octets (last octet of padding indicates how many octets should be ignored)
- X – extension (there is a header extension after the fixed header)
- CC – number of CSRC identifiers that follow
- sequence number – is incremented by one for each RTP data packet sent, and may be used by the receiver to detect packet loss and to restore packet sequence
- timestamp – reflects the sampling instant of the first octet in the RTP data packet
- SSRC – synchronisation source; the source of a stream of RTP packets, in order to make the sources independent upon the network address.

- CSRC – Contributing Sources; source of a stream of RTP packets that has contributed to the combined stream produced by an RTP mixer

Note: to the existing RTP data packet header can be added a RTP header extension.

Although the RTP packet has the timestamp field, this is less used in the analysis; it is an integer, and it is measured in sampling units (depending on the codec used). It is put by the sender and used by the receiver as a reference for the stream playing. The time analysis performed is based on the timestamp of the packet, put by the capturing device, at the monitor.

The following types of parameters can be determined using the RTP header fields and the arrival timestamp of each packet, taken from the packet capture program:

a. delay-related parameters:
- inter-arrival delay – by subtracting the capture timestamps of successive packets
- inter-arrival jitter – by comparing the previous delay with the current one
- one-way delay jitter – by comparing the inter-arrival delay with the sender delay (the interval between sending two sequential packets).

b. packet-accounting parameters
- lost packets and out of order packets – by comparing the expected sequence number with the sequence number of the incoming packet. The lost packets variable is increased, but the presumed lost packets sequence numbers are memorised, in case the packets were not lost, but only misordered.

c. flow speed parameters
- throughput – determined by dividing the actual received number of bytes by the time of the connection

The RTCP packets can be used as an instrument for end-to-end measurements. Their fields provide the values for inter-arrival jitter and lost packets; their structure is also defined in (Schulzrinne et al, 1996), but the header is structured, and too complex to be detailed within this article. RTCP flows perform the following functions:

- to provide feedback on the quality of the data distribution
- to help the receivers to associate and to synchronise multiple data streams from a given participant
- to allow each participant to keep track of all the other participants in the conference
- to convey minimal session control information

The RTCP reports are a very convenient tool for monitoring and they are, as mentioned, currently used in the available products. Nevertheless, the following observations can be made in relation to using RTCP to analyse the flows:

- it runs on UDP and, therefore, it is possible that a number of packets will not arrive, so no data will be available for that period of time.
- it has scalability problems (Rosenberg and Schulzrinne, 1998). The RTCP messages are limited to 5% of the whole traffic. In the case of a many-to-many conference, on

normal behaviour, there would be a low number of RTP messages per-terminal (in order to maintain the 5% limit) (Schulzrinne et al, 1996).
- it returns only end-to-end parameters and, therefore, cannot locate the cause of parameter changes (this problem exists regardless of the conference characteristics)

Note: the analysis is performed on a 'per-flow' basis. Prior to performing the analysis, the incoming packets (from several audio channels) are split into flows (each flow representing a channel). When saying successive packets, we refer to packets belonging to the same flow.

Correlating RTP analysis with RTCP content

By correlating the two sets of parameters, obtained from RTP and RTCP, it is possible to determine whether or not a specific problem (e.g. a high number of lost packets) is caused by a problem which exists in the East sub-network or the West sub-network. Figure 3 presents the captured flows.

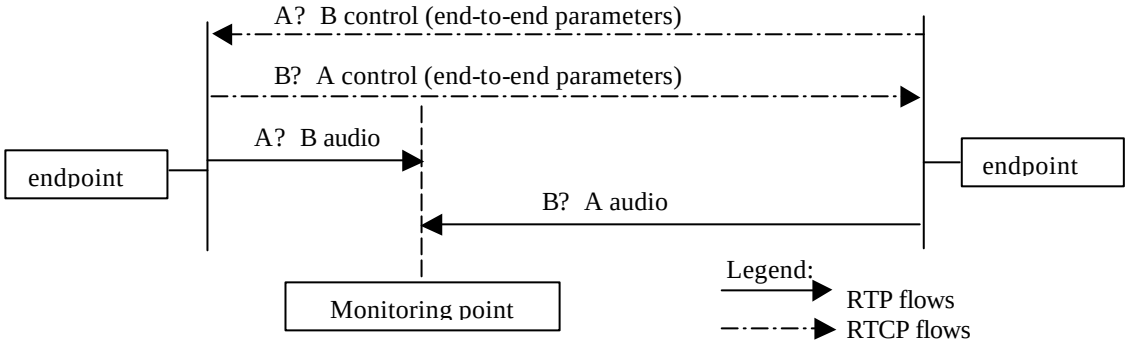


Figure 3: RTP and RTCP flows monitoring

The RTP streams, as captured on the monitoring point, are: A? B (after passing through the West sub-network) and B? A (after passing through the East sub-network). Therefore, by measuring the parameters of these flows, we can determine the performance of the West sub-network (from the A? B flow) and the East sub-network (from the B? A flow).

We have to bear in mind that the A? B direction does not fully characterise the behaviour of the network, as it can be very good for one direction and bad for the other (it does not have to be symmetrical in terms of performance). Meanwhile, as mentioned, RTCP provides the end-to-end parameters, i.e. the performance of the entire A? B and B? A routes, but it has no indication about how these parameters change on the route (i.e. cannot establish where a faulty behaviour of the network determined a change in the values of the parameters).

Putting together the two sets, we obtain parameters for the following segments:

- A? B and B? A, end-to-end – from the RTCP flows
- A? monitoring point and B? monitoring point – from the RTP flows
- monitoring point? B and monitoring point? A – by subtracting the RTP obtained values from RTCP end-to-end parameters.

Therefore, by using both RTP and RTCP, we obtain both the end-to-end and the end-to-monitoring point parameters for the monitored flows.

Implementation

In the first instance, the tcptrace program (Ostermann, 2000) was used within the monitoring module. Tcptrace is an offline analysis program, which uses tcpdump traces as input. Although the program had limited support for UDP (it was able to separate the UDP flows), and no support for RTP, it was considered a useful tool because of its per-flow analysis capabilities. The module was subsequently migrated to ipgrab (Borella, 2000) to reduce the complexity of the program (tcptrace includes a lot of functions, spread over various modules, most of them related with TCP analysis). Most of the analysis (e.g. the distributions), as described in the following section, was performed offline, under Microsoft Excel. As no equipment to simulate several calls was available, the analysis was performed for only a single VoIP call. The module will work for more than one call, but a proper filtration of the output should be added. In addition, the refresh period of the analysis (i.e. each packet) could create computational problems for a high number of flows. A proper solution would be to display the parameters at certain intervals (e.g. every second).

Special attention is given to the marker, payload type and timestamp fields within the RTP header. During a VoIP call, if there is no speech from the user, an endpoint does not send RTP packets. Therefore, when calculating the flow speed and the delay parameters, the silence periods should be ignored. The silence periods can be identified using the marker field: an RTP packet with the marker field set signals the end of a silence period. Also, if the payload characteristics are known (e.g. each RTP packet contains a 30ms frame), the delay between successive packets at the sender can be determined. Thus, by subtracting this value from the inter-arrival delay, we obtain the one-way delay jitter.

Validation

Experimental testbed configuration

A network testbed was constructed in order to validate the proposed method. Figure 4 presents the testbed configuration, which included two networks, connected through a faulty link. The monitoring point is placed on the route, at the exit point (after the router) of one of the networks.

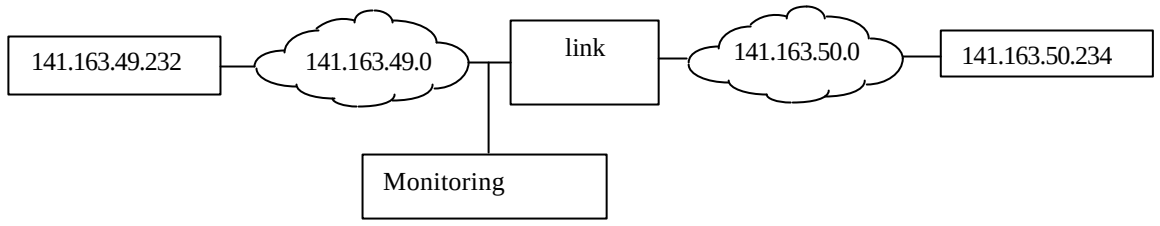


Figure 4: Network testbed configuration

The link is emulated using the NISTNet program (NISTNet, 2000). NISTNet emulates various network problems by forwarding packets, under specific parameters like packet loss, delay or jitter, between two network interface cards, on a Linux station. For our test, we used

the following parameters (symmetric for the two directions): 5% packet loss, 300 ms delay, 25 ms jitter, unlimited bandwidth, normal distribution. The measurements were based on a capture session; number of packets captured: ~20000 (some of them were removed in order to eliminate the transitional behaviour).

The software tools used for generating, capturing and monitoring the VoIP flows were:

- NetMeeting (WinNT) – to establish and run a H.323 VoIP call;
- codec: Microsoft G.723.1, 6400 bits/second, continuous speech;
- tcpdump, ipgrab (Linux) – to capture packets transmitted over the network (between the two VoIP endpoints);
- the analysis module (Linux) – first developed within tcptrace, then transferred to ipgrab, to allow online capturing.

The measurements aim to locate the jitter and the packet loss by dividing the route of the packets, as presented in Figure 3 into sub-network East (network 141.163.49.0), and sub-network West (emulated link and network 141.163.50.0). After obtaining the various parameters, we will try to identify the fault location on the 141.163.50.0 network and link side of the route. In the following paragraphs, we will refer at 141.163.49.232 station as A and at 141.163.50.234 as B.

Results and value comparison

Table 1 presents the following information:

- normal – the normal behaviour, on a network without any loss;
- RTP results – the values determined from the RTP monitoring;
- RTCP results – the values determined from the RTCP parsing.

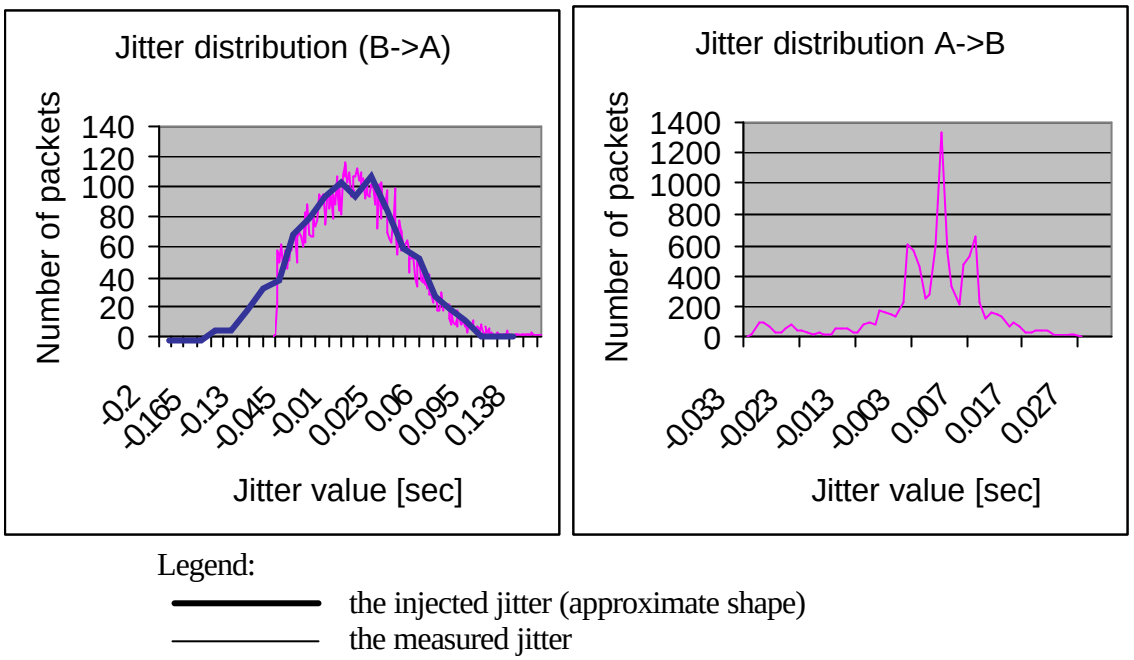
Parameter	normal	RTP results		RTCP results	
		A? B	B? A	A? B	B? A
throughput [bytes/sec]	800	800	760	760	760
packet loss [%]	0	0	5	5	5

Table 1: Throughput and packet loss statistics

A. Throughput and packet loss
The RTCP throughput is determined from the RTCP sender reports, using the 'sender octet count' which indicates how many octets were transmitted since the beginning of the call. The RTCP reports also include report blocks, which give the performance parameters of the senders 'heard' by the emitter of the report. The RTCP packet loss is determined from these report blocks, using the 'cumulative number of packets lost' field.

It can be noticed that the B? A values differs, which indicates a 5% packet loss on that direction, located in the right side of the route. Also, the A? B values indicate that there is no alteration, in term of packet loss, in the left side of the route (the 141.163.49.0 network).

B. Jitter
From the RTP monitoring, the jitter was determined by subtracting the average interarrival delay from the interarrival delay for the current packet. The results are presented in Figure 5.



Legend:

the injected jitter (approximate shape)

the measured jitter

Figure 5: RTP jitter distribution (from RTP monitoring)

Note: In the left graph, the thick line indicates the shape of the average distribution (based on separate measurements on same environment). It can be seen that the measurements are valid.

In the RTCP parsing, the values were extracted from the RTCP report blocks (the 'interarrival jitter' value). The results are presented in Figure 6.

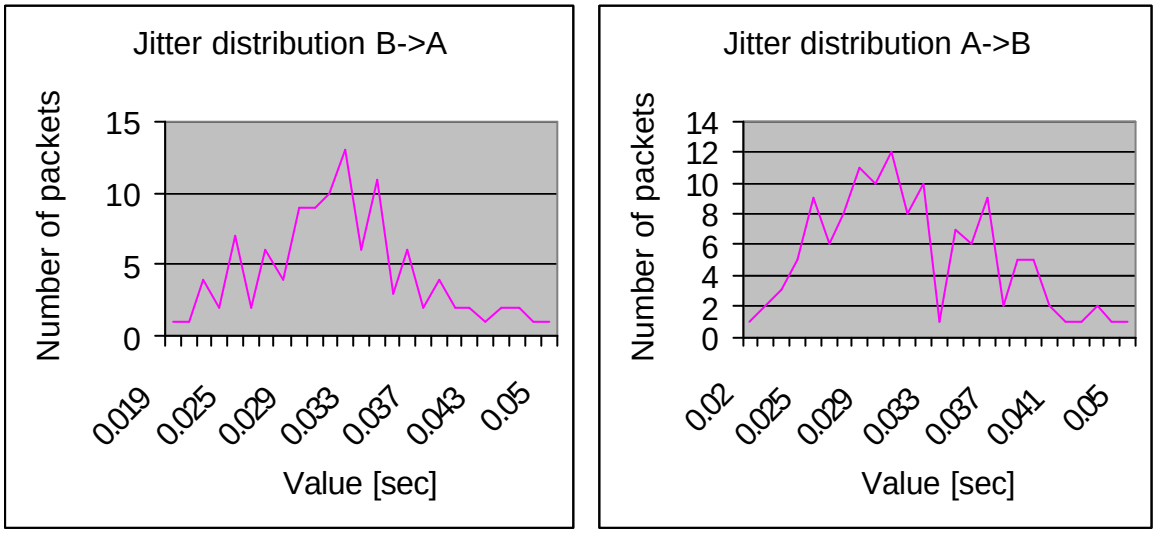


Figure 6: RTP jitter distribution (from RTCP parsing)

As can be seen from Figure 5, the distribution for the B? A flow can be approximated with a Normal (Gaussian) one (the interval (-inf;-0.6) could not be reproduced because of some measurement limitations), while the A? B flow shows no distribution of the jitter. For both of the flows, there is an additional 3 ms jitter, caused by NetMeeting behaviour: although the

packets inter-arrival delay should be constant (60 ms), from time to time, the program transmits a voice packet after 30 ms. The measurement is more accurate for packet loss than jitter because of the errors in the measurement of jitter, as well as because the out-of-order packets were not considered in the analysis.

If we consider the absolute values for the jitter, it results an average value of 28 ms, which, if we extract the 3 ms caused by NetMeeting behaviour, it results the value of the emulated link: 25 ms. As a conclusion, the tool, together with the results analysis, identified the 5% loss and 25 ms jitter generated by the right side of the monitored route.

Although the monitoring tool was built, and these preliminary tests were performed, a full assessment requires further analysis in a real or simulated VoIP environment. Such an environment would include several simultaneous conferences, running between endpoints situated at different locations, over various routes.

Conclusions and further work

This paper has described an off-line method to measure the QoS transport parameters for a H.323 VoIP call from a single point, by non-intrusive monitoring, and we presented a test performed in order to validate our method. The jitter and packet loss analysis seems promising, but further work is required to determine, monitor and analyse the other parameters. Also, a specific change in the performance parameters group can be related with a specific network event (e.g. a congested router). Therefore, analysis of the dynamics of the calculated parameters is required.

There are also other parameters still to be measured. In measurement systems for POTS (Plain Old Telephone Systems), a useful parameter for the call performance is the round trip time (RTT) delay (i.e. the time needed by a signal to go from one end to the other and then back). There is no direct possibility to determine such a parameter for H.323 calls because the standard is built for multicast conferences (multi-to-multi conferences), and so it does not include mechanisms for single end-to-end connection; the flows between the endpoints do not run in pairs, there is no correlation between them (they run independently). There are several methods to determine RTT for VoIP calls:

- Using the setup and control messages; they run on TCP, and the values obtained might differ from the (theoretical) ones for UDP
- Using RTCPs' 'delay since last source report(SR)' field.
- Correlating the RTP and RTCP flows. The RTCP packets include a 'extended highest sequence number received' field. If the value of this field is correlated with the sequence number of the sender, together with its timestamp, the RTT can be measured.

As future work, we aim to:

- refine the described method in order to cover all the possible situations; e.g.: due to method limitation, we were not able to identify correctly jitter higher than the inter-arrival time,
- determine a good estimate for RTT, based on the RTCP reports,
- advance the correlation of RTP and RTCP flows in order to narrow the region of fault location from East/West network down to a link or a sub-network,

- investigate, using intelligent analysis methods, if the traffic performance parameters at one moment can give an estimate for the future level of performance

The method presented, together with the additional objectives above, aims to achieve the perfect monitoring approach, which has to be single point, non-intrusive, measures all the performance parameters, fully locates the source of network degradation, and predicts the future behaviour of the network. By doing this, we can determine if the IP network offers, currently as well as in the future, to the IP telephony users the quality they require, and, if not, where the problem resides.

References

Bacher D. (1996), 'rtpmon: A Third-Party RTCP Monitor', ACM Multimedia '96.

Borella M. (2000), 'ipgrab homepage', http://home.xnet.com/~cathmike/MSB/Software/index.html.

Caceres R., Danzig P.B., Jamin S., and Mitzel D.J. (1991), 'Characteristics of Wide-Area TCP/IP Conversations', Proceedings of ACM SIGCOMM '91.

Ferrari D., Banerjea A., and Zhang H. (1994), 'Network Support for Multimedia – A Discussion of the Tenet Approach', Computer Networks and ISDN Systems, December 1994.

ITU. (1998), 'Packet based multimedia communication systems', H.323 ITU Recommendation, February 1998.

Nistnet. (2000), 'The NIST Net home page', http://snad.ncsl.nist.gov/itg/nistnet/index.html

Ostermann S. (2000), 'tcptrace homepage', http://jarok.cs.ohiou.edu/software/tcptrace/tcptrace.html.

Schulzrinne H., Casner S., Frederick R., and Jacobson V. (1996), RFC 1889 - 'RTP – A Transport Protocol for Real-Time Applications', RFC depository, January 1996.

Crawley E., Nair R., Rajagopalan B., and Sandick H. (1998), RFC 2386 - 'A Framework for QoS-based Routing', RFC depository, August 1998.

Nichols K., Blake S., Baker F., and Black D. (1998), RFC 2474 - 'Differentiated Services Field', RFC depository, December 1998.

Rosenberg J. and Schulzrinne H. (1998), 'Timer Reconsideration for Enhanced RTP Scalability', Proceedings of IEEE Infocom 1998, March 29 - April 2 1998.

Stiller B. (1997), 'Quality of Service Issues in Networking Environments', internal report, http://www.cl.cam.ac.uk/ftp/papers/reports/TR380-bs201-qos_isues.ps.gz, September 1995.

Thompson K., Miller G.J., and Wilder R. (1997), 'Wide-Area Internet Traffic Patterns and Characteristics', IEEE network, November-December 1997.