Reactive Mechanisms for Recovering Audio Performance in Multimedia Conferencing Over Packet Switched Networks

Hani ElGebaly, Intel Architecture Labs, Intel Corporation

Index words: RTCP, H.323, congestion, multimedia, jitter, packet loss

ABSTRACT

The impact of multimedia traffic on the performance of networking applications is significant because the vast majority of currently available tools send data at a rate that does not depend on the state of the network. This is unlike rate-controlled applications, such as Transport Control Protocol-based (TCP) applications, which adjust their output rate and bandwidth requirements according to the state of the network. Audio is the most important component in a multimedia session. It is thus important to grant audio the maximum attention even if it necessitates sacrificing the throughput (but not the quality of service) of other less important applications. An example of such a sacrifice is to reduce the video bit-rate if the audio quality drops. Reducing the bit-rate may or may not be observed by the user. A video source may adapt to this change by changing the target frame rate if the drop in bit-rate is significant.

In this paper we address feedback algorithms for dealing with audio loss problems. We propose a modification to a Real-time Control Protocol-based (RTCP) congestion control algorithm that is sensitive to audio performance. We analyze the RTCP-based congestion control technique and present its limitations. We then uncover a relationship between audio jitter and audio packet loss. Finally, we present a novel scheme for predicting loss using jitter information.

INTRODUCTION

Advances in computer and communication technology have stimulated the integration of digital audio and video with computing. This integration led to the development of multimedia applications such as video conferencing and multimedia collaboration. Many of these applications are targeted to run over packet-switched networks such as the Internet. Packet-switched networks with non-guaranteed quality of service do not seem suitable for real-time traffic such as multimedia conferencing.

Internet switches do not treat network traffic in a fair way. Packets are routed independently across shared routers and switches. These switches do not pay attention to the type of packet, packet loss, or latency sensitivity. As parts of the Internet become heavily loaded, congestion can occur. Congestion may led to buffer overflow and packet loss. It may also lead to packet delay as packets take longer to process. Latency may seem acceptable for some applications such as e-mail and file transfer. For real-time applications, data becomes obsolete if they do not arrive in time. In addition, real-time applications can be quite sensitive to loss. Furthermore, since packets are routed independently across shared routers and switches, transit times may vary significantly. Variation in transit delay is called jitter, and jitter is very disturbing to real-time applications, especially to audio playback. It significantly reduces speech intelligibility to the ear and causes choppiness and breakup in the stream. Consequently, real-time applications deliver poor quality during periods of congestion over shared bandwidth networks such as the Internet.

Several researchers have addressed these Internet problems and have proposed infrastructure enhancements or modifications to the current Internet TCP/IP-based infrastructure.

Infrastructure enhancements call for new technologies, for example, fiber optics [2], better and faster switch architectures [1], and new protocol layers such as B-ISDN [7] ATM [12], etc.

Infrastructure modifications call for implementing congestion control mechanisms in the network routers that reward rate-controlled applications and punish aggressive ones. These modifications require that router software and the current infrastructure of the Internet be modified. The purpose of these modifications is to provide ways of enforcing resource reservation, quality of service bounds, and recognition of real-time traffic. For example, the RSVP [14] protocol maintains a specific quality of service for participants in a communication session. In this model all routers along the data path keep a record of all current service requests and check all packets to see if they need special treatment. Another example is the Internet Protocol (IP) type of service (TOS) byte that was intended to express eight precedence levels of service (in an increasing precedence order). Unfortunately, this service is underutilized by implementers. The IETF redefined those TOS bits in the IP protocol as a differentiated services field [3]. The differentiated services field will contain a six-bit flag assigned by the server at the edge of the network or by a classifier built into a router to grant the packet a class of service based on its priority.

Other solutions for the congestion problem include implementing TCP-friendly rate control algorithms, jitter compensation at the receiver, and forward error correction fault-tolerant algorithms.

Undoubtedly new technologies will evolve over the years. The deployment of these technologies may take some time, but eventually will make it to the implementation phase. Indeed many of these technologies have already been running for years over private networks exhibiting outstanding performance. Yet, solutions for current networks are essential since they will persist for a while as a vital communication media for many end users.

In this paper, we discuss two TCP-friendly rate control algorithms that save the audio quality in a multimedia connection. Both algorithms are based on reactive mechanisms used to recover media flow performance degradation caused by the effects of shared bandwidth traffic. We present an overview of feedback mechanisms based on the Real-time Control Protocol (RTCP). We uncover the limitations of RTCP feedback on applications connected to the Internet through narrow bandwidth pipes. We propose an alternative approach to the RTCP feedback algorithm that predicts and prevents the loss of audio packets before it occurs, based on local computation of incoming audio jitter. These approaches improve the audio performance significantly in multimedia conferencing sessions.

LOCAL AND REMOTE SCHEMES FOR CONGESTION HANDLING

The poor performance of multimedia conferencing over the Internet can be attributed to two main factors: local- and remote-induced effects. Local effects are induced by bandwidth sharing between different media components, operating system limitations, or poor design. Remote effects include all Internet-related problems such as unfairness, non-guaranteed quality of service, congestion, etc.

Consequently, solutions to overall audio latency and jitter in a conferencing session are either local to the host (local schemes) or based on feedback from the remote conferencing peer (remote schemes), or a combination of both.

Local schemes address issues such as careful scheduling of video and audio traffic at the host before media components reach the shared bandwidth network layer. The objective of these schemes is to prevent video from oversubscribing to the available bandwidth, thereby degrading the performance of the audio stream.

Remote schemes rely on statistics such as loss, jitter, latency, timestamps, etc. exchanged between network terminal peers. A terminal, upon receiving these statistics, will decide to adjust the cumulative throughput of one or more media channels in order to make up for loss, latency, or jitter symptoms. This class of feedback scheme is called Remote Feedback Local Control (RFLC). An example of this kind of feedback is the RTCP reports used in H.323 [9] conferencing and general multicast streaming. Alternatively, terminals can compute these statistics based on the actual stream of media data and provide Flow Control commands to remote terminals to limit throughput of the degrading media type. In this paper, we discuss this class of feedback scheme known as Local Feedback Remote Control (LFRC). Our objective is to devise a scheme that can predict the occurrence of audio loss and take appropriate actions to prevent it.

MAIN CAUSES OF POOR AUDIO PERFORMANCE

There are mainly three reasons for poor audio performance: packet loss, delay, and jitter. Packets may be lost in the network and never arrive at the receiver. Audio loss is manifested as lost phonemes in a word or even lost words in a sentence. It can also be manifested as breakage in the continuity of audio playback. Packet latency leads to audio frames arriving late. Latency disturbs speech continuity and interactivity. Packet inter-arrival jitter causes choppiness of audio playback at the receiver.

Packets get lost due to buffer overflow or due to bit errors. The probability of bit errors is very low on most packet-switched wired networks. Hence, loss is mostly induced by buffer overflow [11]. Buffer overflow can happen on a congested link (Internet intermediate switch) or at the network interface of the workstation. The network interface loss on low speed links can be due to a narrow bandwidth pipe supplied by the Internet service provider. Controlling the bit-rate of the media stream can alleviate the loss problem.

Packets are delayed and jittered because of long waiting time in Internet intermediate switch queues, long routes, and heterogeneous link speeds. Unless Internet infrastructure substitutes the current best effort routing algorithm with another fair and multimedia-aware algorithm, there is little hope for conquering the delay problem effectively in today's Internet.

Packet Loss and Latency Contributors

Modem Connection Latency

The latency contributed by a modem connection is composed of the per-byte transmission time (modem bandwidth) and a fixed software and hardware overhead.

The transmission time depends on the transmission rate of the modem. The fixed hardware and software overhead varies between different architectures and platforms. One delay factor, which is attributed to that overhead, is grouping of data. There is a modem wait time as the modem tries to group data into blocks and perform compression and automatic error correction. To get effective compression and error correction, modems must work on large blocks of data. This means that characters must be buffered until a sufficient block is built for the modem to work on efficiently. Data are not sent until processed by the modem's compression/error correction engine. This adds to the latency of data as they pass through the modem. In addition, the modem does not know the kind of data being sent, and consequently cannot use the best data-specific compression algorithms. For example, multimedia data are usually compressed before they pass through modems. Modem compression in this case is futile. In fact, compression may significantly affect the timeliness of the latency-sensitive data transferred through the modem.

For a typical modem link, the latency due to modem software and hardware overhead is usually about 100ms. The transmission time of 10 characters over a 33kbit/sec modem link time would theoretically be 80 bits / 33000 bits per second = 2.4ms. The actual time taken because of the compression overhead is 102.4ms. This is because of the 100ms latency introduced by the modems at each end of the link.

To enable small chunks transfer, there is a timeout value before the modem starts processing a data block. Using this timeout value, modems can avoid waiting indefinitely for the large block and so avoid causing huge delays to the peer. Hence, modem hardware and software overhead is a significant contributor to latency when connected to the Internet via modems through ISPs.

Internet Service Providers Bottleneck

The Internet service provider is the means by which the home user and many businesses hook up to the Internet. Users connect to their ISPs usually over Plain Old Telephone Service (POTS) lines via modems. An Internet provider in turn must connect to another wholesale Internet provider. The connection of an ISP to the wholesale provider is called the ISP pipe. Most ISPs anticipate relatively low bandwidth activity from their users such as Web browsing, reading information, etc. Multimedia conferencing is rarely taken into account. For example, Table 1 shows how an ISP pipe size may map to the number of ISP users [5]. Table 1 does not take into account if users are engaged in multimedia conferencing sessions. In the future, however, ISPs will have to satisfy the requirements of their conferencing clients.

Typically, each conferencing client logged on to an ISP via a 33.6kbps modem will chew up at least 8-10kbps for audio and 16-22kbps for video of the available bandwidth. Note that both numbers almost add up to the dialup connection bitrate available. This leads to a significant amount of latency and loss during the conferencing session. Further, most ISPs are oriented to downloading (down-streaming). Their down-streaming pipe is usually wider than their up-streaming pipe. Hence, packet loss is more significant when a conferencing client is sending media data upstream.

Pipeline Size	Number of Simultaneous Dial-up users
28.8K modem connection	3
56K DSO leased line	7
64K ISDN connection	8
128K ISDN connection	15
256K Fractional T1	40
512K Fractional T1	100+
Full T1	300+

Table 1: ISP pipeline size versus number of users

An overview of the streaming protocol that was used for this study is given in the next section.

H.323, RTP, AND IP TELEPHONY

IP telephony has become an important driver for packet-based communications. The H.323 recommendation developed by the ITU has provided a good basis for establishing a universal IP voice and multimedia communication in large, connected networks [8]. By using Q.931 as its basis for establishing a connection, H.323 allows for relatively easy bridging to the public switched telephone networks (PSTN) and circuit-based phones. The required voice codec of G.711 also allows for easy connections to the legacy networks of telephones. As the standard evolved, additions and extensions made it more suitable for IP telephony. H.323 has been declared the standard for Voice over IP communication by international standards' bodies such as the European Telecommunication Standard Institute (ETSI).

H.323, similar to a few other protocols, uses the Real-time Protocol (RTP) for data streaming. RTP is a real-time protocol for multiplexing media components in a media stream connection. It does not offer any form of reliability nor does it ensure real-time delivery. The protocol is real-time in the sense that it provides functionality suited for carrying real-time content, e.g., a timestamp and control mechanisms for synchronizing different streams with different timing properties.

RTP assigns timestamps for multimedia packets such that the timing order is restored at playback side. It also assigns sequence numbers in order to be able to detect losses. The RTP packet header size is 12 bytes. It carries information about payload type, source identifier, time stamp, and sequence number. Although RTP is an unreliable protocol, it provides hooks for adding reliability through fault tolerance, feedback reactive algorithms, and other reliability schemes as discussed in the following sections.

Along with RTP there is another protocol that is responsible for packaging information about connection quality statistics. This protocol is termed the Real Time Control Protocol and is discussed next.

REAL TIME CONTROL PROTOCOL – RTCP

RTCP is an associated protocol with RTP designed to handle the delivery monitoring service of the RTP protocol. It is based on periodic transmission of control packets, collected from all participants of a session, to all other participants. It provides feedback on the quality of media packets including timing information. All RTP senders generate a sender report. A sender report contains information useful for media synchronization as well as cumulative counters for packets and bytes sent. This information allows receivers to estimate the sender's data rate. Receivers generate receive reports for all video and audio sources they have heard from. These reports contain information on the highest sequence number received, the number of packets lost, a measure of the inter-arrival jitter, and the timestamps needed to compute the round-trip delay between sender and receiver issuing the report.

RTCP packets also contain a unique identification for their original sender via a source description (SDES) packet. This packet may contain user name, canonical name, e-mail, etc.

In the next section we propose an algorithm based on RTCP reports feedback that is a modification of Busse's algorithm [1].

RTCP FEEDBACK CONGESTION TECHNIQUE

Busse et al. [6] introduced a dynamic quality of service (QoS) control mechanism for multimedia applications based on RTCP feedback. Their technique is based on receiver reports delivered from receiving end applications to the source. This new algorithm is a modification of this approach. It is based on the use of an audio receiver report to adjust video bandwidth in a multimedia session between two peers. The objective is to provide audio with enough bandwidth to function with an acceptable QoS by sacrificing some of the video bandwidth.

The network is divided into distinct congestion states. Each congestion state is determined by the amount of audio packet loss experienced in the previous determined interval. The interval is the inter-arrival time between RTCP reports packets. The number of audio packets lost determines whether the network is in the unloaded, loaded, or congested state as depicted in Figure 1.

Assume that two H.323 terminals are engaged in a H.323 conferencing session. Both terminals stream audio and video to each other on separate UDP connections. Both terminals also exchange sender and receiver RTCP reports on separate UDP connections for both audio and video. The conferencing manager at each terminal, on receiving an audio RTCP receiver report (ARR) from a remote peer, does the following:

- Analyzes the statistics reported by the ARR for loss.
- Determines the network congestion state. A network can be in an unloaded, loaded, or congested state. The loss value obtained from the ARR determines the current network congestion state. This value is used to decide whether to increase, hold, or decrease the video bit-rate of the terminal.

- Adjusts video bit-rate according to the network state analysis. The user has the freedom to specify the minimum and maximum video bandwidth allowed. The user also has the freedom to specify audio parameters that determine the network's states and can adjust the video bit-rate accordingly.

Parameter values for determining the current state of the network are shown in Figure 1. λc is the lowest audio packet loss value beyond which the network becomes congested and video bit-rate has to decrease by δc. λu is the highest audio packet loss value below which the network is lightly loaded (or unloaded) and video bit-rate will increase by δu. During the loaded state, the video bit-rate is unchanged. The choice of the parameter values is subjective and may depend on the current condition of the network.

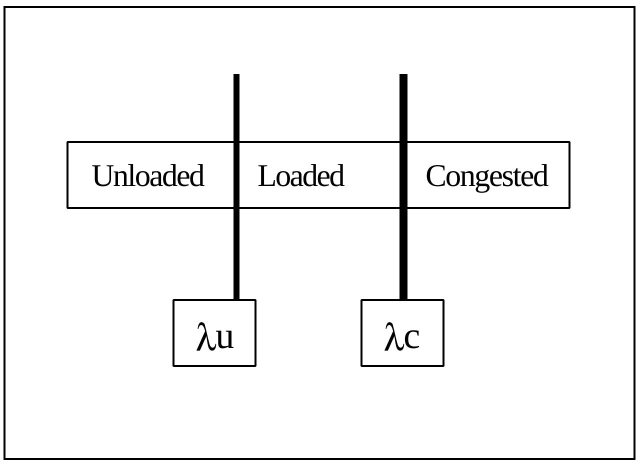

Figure 1: Different network states

We implemented the above algorithm on two Intel H.323-based Video Phone terminals. Both terminals were connected over the Internet via different ISPs. The maximum video bandwidth was set to 25 kb/s. Both terminals used G.723.1 as their preferred audio codec and H.263 as their preferred video codec. An audio interview file that lasted for approximately ten minutes was used as the audio input sample. We chose δc = 1 kb/s, δu = 0.5 kb/s, λc = 45 packets, and λu = 30 packets. The experiment was carried out during a known Internet rush hour to simulate the worst congestion scenario. Both terminals were running full-screen large format video (CIF) and low bit-rate audio.

Figure 2 shows the relationship between video bandwidth and audio packet loss as the session proceeds. It is observed that as audio loss increases beyond the congested state, video bit-rate decreases quickly, and consequently audio loss starts to improve. Video bit-rate starts to rise again whenever audio loss falls below the loaded state. The choice of parameters that determines the state of the network is user dependent. Our choice was based on statistics collected after running multiple sessions in a similar environment.

The algorithm increases the video bit-rate as long as the audio loss is below the network threshold and the network is in the unloaded state as depicted in Figure 1. Once the audio loss starts to increase, exceeding the network loaded state and entering the congested state, the algorithm reacts by decreasing the video bit-rate pulling the network back to the loaded state. Note that the rate of increase (slope of the curve) of the video bit-rate is slower than the decrease rate in order to gain more control over the audio congestion problem. The algorithm will maintain the video bit-rate at its current level as long as the loss rate is within the limits of the loaded network state.

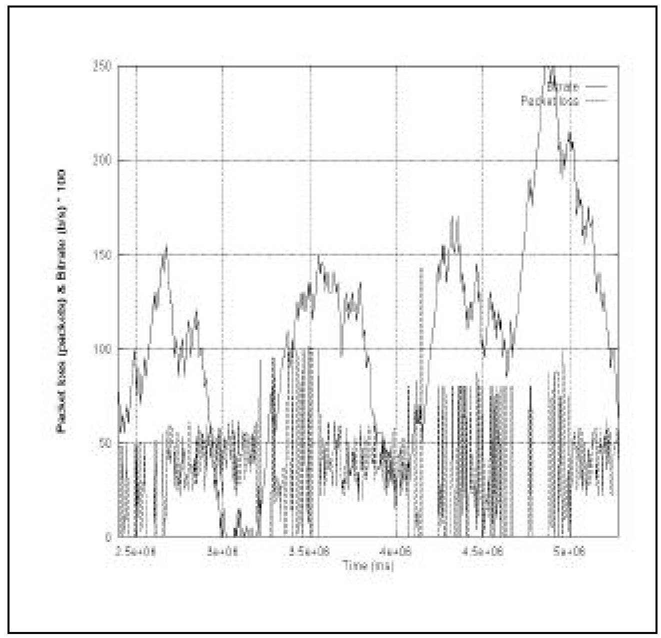

Figure 2: Video Bit-rate versus audio loss

There are several disadvantages to this approach. These disadvantages are discussed in the following section.

RTCP FEEDBACK LIMITATIONS

In order to keep RTCP overhead to a minimum, thereby better utilizing data packets, RTCP packet transmissions are limited. In general, no more than five percent of the total bandwidth of a media stream can be allocated to RTCP functionality [1]. On slow links with limited bandwidth, the inter-arrival time between successive RTCP receive reports may exceed seven seconds. This interval is quite long and may lead to slower response of adaptive schemes to network conditions. In other words, more audio loss than expected may occur before the terminal reacts by decreasing the video bit-rate. A more responsive approach is required to effectively control the flow of all media streams. Hence a new feedback algorithm is required that does not wait for late statistics from the network and responds promptly to irregularities in the network.

Another limitation is the lack of criteria to determine accurate estimations of network state parameters. Internet home users may have relaxed requirements for the parameters; however, these parameters may not be acceptable to business users. The user has to rely on statistics for each particular case.

A third limitation to the RTCP feedback approach is that a possible QoS oscillation may occur where video bit-rate and audio loss follow a sinusoidal waveform. This phenomenon will make audio conversation unintelligible because of large peak losses. Video, on the other hand, will also suffer from poor performance because of reduced bit-rate. Long inter-arrival intervals between reports together with choice of control parameters contribute to this phenomenon.

A fourth drawback to the RTCP feedback approach is related to its periodic nature. Congestion or packet loss is a sporadic event; hence, it has to be treated by another non-periodic event. In fact, network congestion caused by audio loss, particularly in real-time applications, is short-lived and non-correlated [4]. Thus, reaction to loss should be prompt and instantaneous. In addition, waiting until loss happens and reacting afterwards may not be the best approach. A better approach is to provide a means to predict loss that informs the media terminal to slow down in order to avoid loss from happening. This approach is discussed in the following section.

RELATIONSHIP BETWEEN PACKET LOSS AND JITTER

An experiment was conducted using similar input and network conditions to those used in the case of the RTCP feedback experiment. We measured audio packet loss, inter-arrival time between packets, and inter-arrival jitter at the receiver. -The packet loss is measured as the difference between the sequence numbers of packets arriving at the receiver. -If the difference is greater than one, then loss occurred. —The inter-arrival jitter is measured as the difference in packet spacing at the receiver compared to the sender for a pair of packets as suggested in [1]. This is equivalent to the difference in the “relative transit time” for the two packets. The relative transit time is the difference between a packet’s RTP timestamp and the receiver’s clock at the time of arrival. This value is measured in milliseconds.

Figure 3 shows a plot of the inter-arrival time according to the receiver clock and audio packet loss, both against time. Figure 4 shows an extension in the time domain for the same plot. The audio packets used in this experiment carry four audio frames per packet. The frame length for the G.723.1 codec is 30ms. Hence, the packetization time is almost 120ms. Further, the audio session that was used for this experiment had very small (almost negligible) silence periods. Both Figures 3 and 4 show that change in the inter-arrival time of the audio packet was followed by a jump in packet loss. It is observed from the plots that the peaks of packet loss often occurred shortly after an abrupt change in the inter-arrival time of previous packets. Smaller changes in the packet inter-arrival times were followed by no or just small packet losses. The more significant the change in the inter-arrival time, the higher the loss value as shown in the figures. This was a motivation to pursue this observation further and find out if there is any relationship between audio inter-arrival jitter and packet loss.

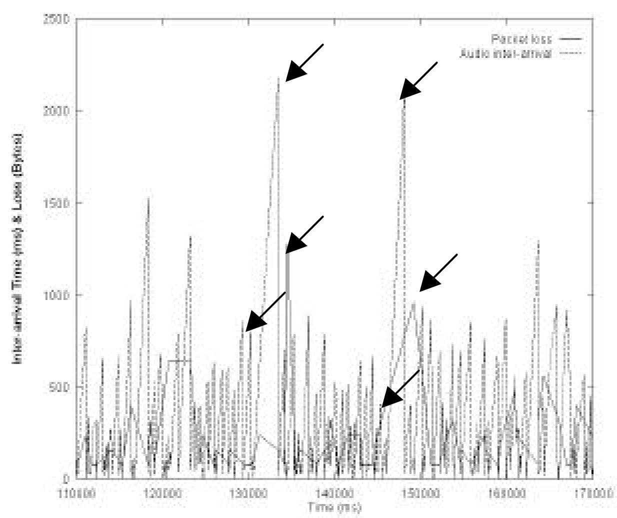

Figure 3: Audio receiver inter-arrival time and packet loss

We computed the audio inter-arrival jitter at the receiver. The audio jitter is computed as the rate of change of the inter-arrival time. The jitter value can be positive or negative.

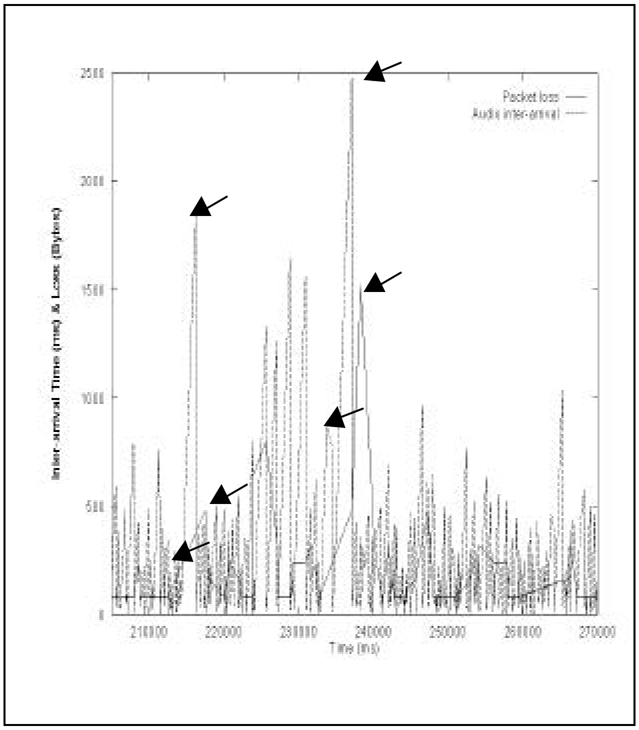

Figure 4: Audio inter-arrival time and packet loss

Figure 5 shows a plot of the relationship between the inter-arrival jitter and packet loss. Figure 6 shows an extension of the relationship in the time domain. It is observed that the negative audio jitter peaks are of larger amplitude and occur more frequently. The reason is that the inter-arrival time between packets at the receiver is often larger than that on the source since the source is usually controlled by a constant bit-rate regulator. It is also observed that the jitter at the receiver is of positive value. The reason is that the packet inter-arrival time at the receiver can be lower than that of the source. Since the inter-arrival value at the sender is usually maintained at a constant value (in our case 120 ms, since we pack four frames per audio packet), the peak positive value of the jitter is not expected to exceed this value. There may be a few exceptions to this rule as shown in Figure 5. One exception to this rule is when an error occurs at the source, causing it to violate the constant bit-rate rule. Another exception is in the case of silence periods at the source where packet inter-arrival time may be longer. These exceptions explain the very few occurrences of positive high values for the jitter.

Figure 5 shows small overshoots for packet loss following abrupt rises in the audio jitter of previous audio packets. Figure 6 shows increases for packet loss overshoots as the jitter magnitude increases for the previous audio packets. Both curves showed consistent results with packet loss moving up and down based on the amount of jitter of the previous packets. More experiments are needed in order to make a generalization for the loss-jitter relationship, yet the obtained results provide a good basis for the introduction of a loss prediction algorithm. Our observation from Figures 5 and 6 suggests that abrupt rises in jitter value may result in packet losses after a short interval of time. The relationship between audio packet loss and jitter implies that jitter computation at the receiver can predict the possibility of packet loss in the near future. Hence, by carefully monitoring inter-arrival time at the receiver, it is possible to reduce packet loss by warning the sender.

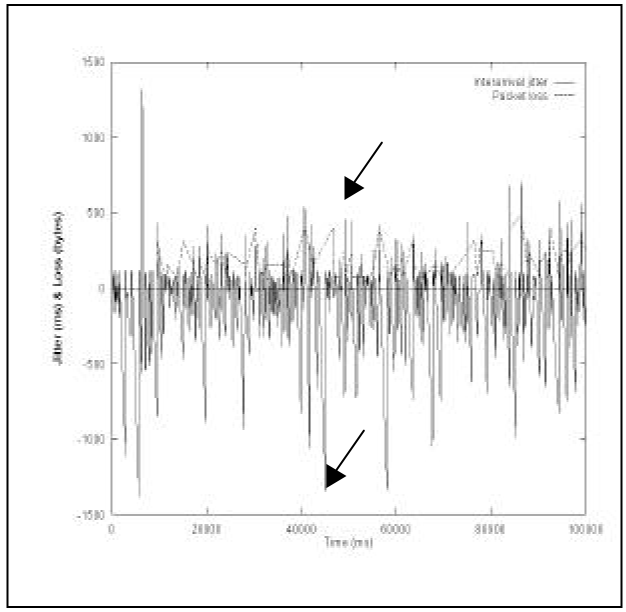

Figure 5: Inter-arrival jitter and packet loss

H.323 defines procedures for controlling the bit-rate of the multimedia streams in the conference using the H.245 [10] Flow Control command. The Flow Control command can modify the bit-rate of the channel stream or of the whole multiplex of streams. A terminal may send this command to restrict the bit-rate sent by the far-end terminal. According to H.323 recommendations, a terminal that receives this command shall comply with it. Using the H.245 Flow Control command and the relationship between audio jitter and loss, we propose an algorithm for audio loss prediction and control as outlined in the following section.

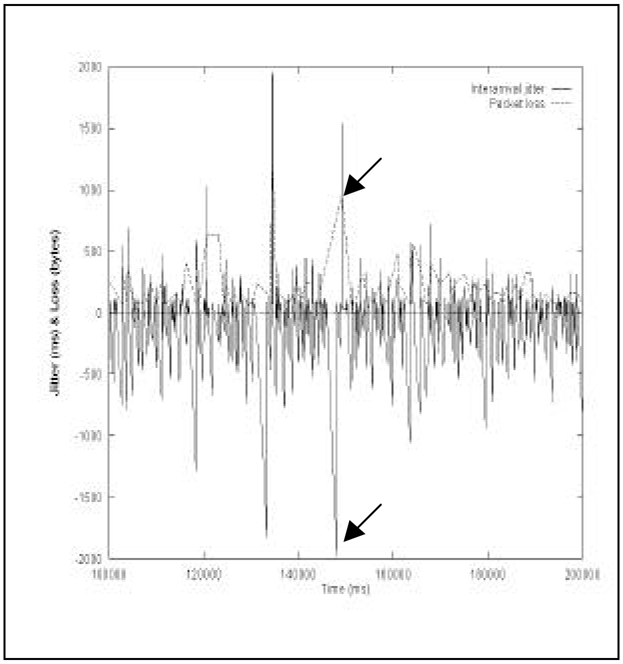

Figure 6: Inter-arrival jitter and packet loss

THE LOSS PREDICTION ALGORITHM

The previous measurements were performed for audio inter-arrival jitter and loss only, and it is yet to be seen if the same relationship applies to other real-time media components such as video. The audio loss-jitter relationship motivated the development of another algorithm that can predict the occurrence of packet loss. This algorithm works as follows:

- Compute packet inter-arrival jitter J_i for each audio packet i at the receiver.
- If the value of J_i exceeds a maximum value J_{max}, the network is in the congested state and loss may occur. Send a Flow Control command to the sender to limit the video bit-rate.
- If the value of J_i drops below a minimum value J_{min}, the network is unloaded, and the video bit-rate can be restored to default value. Send a Flow Control command to the sender to increase the video bit-rate.
- If the value of J_i falls between J_{max} and J_{min}, the video bit-rate will not be adjusted.

This algorithm controls the sender based on the loss predictability of the receiver. The algorithm is considered more binding than the RTCP-based algorithms since it enforces a certain behavior (increase or decrease of video bit-rate) instead of just providing feedback on the connection.

A simulation was run using the data obtained from the audio jitter-loss relationship experiment to compute the number of jitter warnings and the corresponding loss occurrences. The purpose of this simulation was to verify the loss prediction algorithm described above.

Figure 7 shows the number of jitter warnings and the percentage of unpredicted loss on the y axis both as a function in J_{max} (on the x axis). The jitter warnings are normalized against the maximum number of warnings that occurred during the experiments, and J_{max} is measured in units of the standard deviation of the inter-arrival jitter.

Using a high J_{max} value, the loss prediction algorithm was capable of predicting almost 96% of lost audio packets. As the threshold value, J_{max}, decreases, the packet loss value starts to rise.

It is observed from Figure 7 that a good suggested value for J_{max} that provides a compromise between the number of generated jitter warnings and packet loss is approximately 1.8 * jitter standard deviation. This J_{max} value provides prediction closure for a majority of packet loss.

It is possible to optimize the number of generated Flow Control commands by following certain heuristics. One possible heuristic is to limit the number of generated jitter Flow Control packets within a certain interval of time provided that the last computed jitter was not preempted. Preemption means that a new jitter value was computed with a larger absolute value than the last computed jitter value that caused the video Flow Control command to occur. Likewise, stabilization in the jitter computation at the receiver can prompt the receiver to issue Flow Control commands to raise the video flow rate.

This algorithm is more responsive than the RTCP-based feedback algorithm. It is also possible to implement using standard protocol messages such as the H.245 Flow Control command. The algorithm is also more efficient than the RTCP algorithm since flow control is generated on demand as opposed to periodically generating large statistics' packets. Further, this algorithm is based on local feedback and remote control (LFRC) as opposed to remote feedback and local control (RFLC), for example, RTCP-based algorithms. LFRC is more effective when terminals from different manufacturers are interoperating, since remote control is more binding than remote feedback. The reason is that terminals are required to comply to commands issued from other terminals but can ignore indications (such as RTCP reports). To be more specific, terminals can ignore RTCP reports and choose to not adjust their flow rate based on these reports, while they are obliged to comply with a command such as the H.245 Flow Control. In this way a terminal can effectively control the quality of the stream and the rate of the traffic.

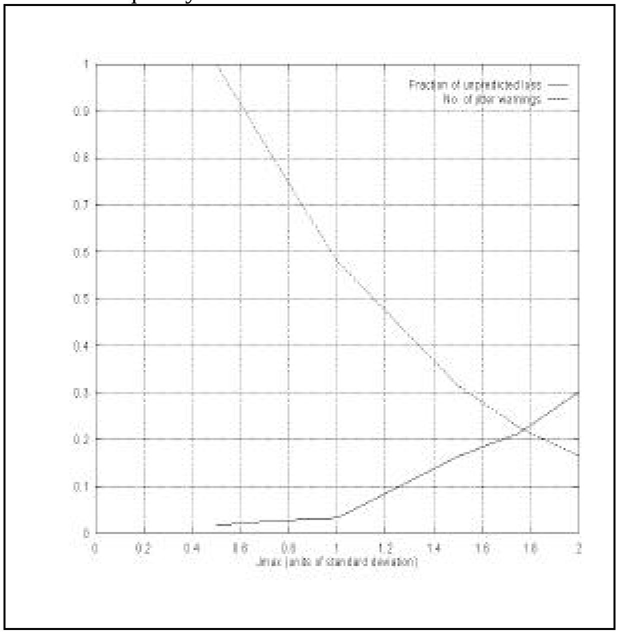

Figure 7: Loss, jitter indications, and J_{max}

CONCLUSION

Multimedia conferencing applications can be rate controlled over the Internet by continuously providing feedback about its performance over the network. Typical useful statistics include packet loss, delay, jitter, and timestamps for media synchronization. RTCP is an associated protocol with RTP designed to handle the delivery of the monitoring service of the RTP protocol. It provides periodic feedback on the quality of media packets including timing information.

We introduced an RTCP-based feedback algorithm that recovers from audio loss by controlling video bit-rate. The algorithm adapted well to network conditions; however, the response was rather slow and tardy since the feedback cycle is larger than six seconds.

We presented the limitation of RTCP-based algorithms for bit-rate control. We then attempted to discover a relationship between audio loss and packet inter-arrival jitter. We computed the jitter and the packet loss at the receiver and plotted their relationship with time. It was discovered that most of the significant packet loss was preceded by abrupt changes in packet inter-arrival time or jitter. We proposed a new algorithm that can predict audio loss based on jitter computation at the receiver. We evaluated the performance of the algorithm to assess its effectiveness. The loss prediction algorithm is more responsive than RTCP-based feedback algorithms. In addition, the prediction algorithm is enforcing control as opposed to RTCP-based algorithms that only provide feedback indications.

REFERENCES

[1] Aravind R., "Image and Video Coding Standards," *AT&T Technical Journal, Vol. 72 No. 1*, January/February 1993, pp. 67-89.

[2] Ballart R., "SONET: Now It's the Standard Optical Network," *IEEE Communications Magazine*, pp. 8-15, March 1989.

[3] Bernet Y., et al, "A Framework for Differentiated Services," *draft-ietf-diffserv-framework-02.txt*, February, 1999.

[4] Bolot J., Vega-Garcia A., "The Case for FEC-Based Error Control for Packet Audio in the Internet," in *ACM Multimedia Systems*, 1998.

[5] Burke C., Busch J., *How to Build an Internet Service Company*, systems 2000+ Publishing, 1996.

[6] Busse I., Deffner B., Schulzrinne H., "Dynamic QoS Control of Multimedia Applications based on RTP," *Computer Communications*, January, 1996.

[7] CCITT, "Recommendations on BISDN," *Geneve*, January 1990.

[8] ElGebaly, H., Toga, J., "Demystifying Multimedia Conferencing Over the Internet using H.323 Standard," *Intel Technology Journal*, Q2, 1998.

[9] ITU-T Recommendation H.323 (1996), "Terminal for Low Bit-Rate Multimedia Communication Over Non-Guaranteed Bandwidth Networks."

[10] ITU-T Recommendation H245 (1995), "Control Protocol for Multimedia Communication."

[11] Jacobson V., "Congestion Avoidance and Control," *ACM Computer Communication Review, Vol. 18*, pp. 314-329, August, 1988.

[12] Le boudec J-Y., "The Asynchronous Transfer Mode: A Tutorial," *Computer Networks and ISDN Systems, No. 24*, pp. 279-309, 1992.

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

[14] Shang, L., Deering, S., Estrin, D., Shenker, S., "RSVP – A New Resource ReSerVation Protocol," *IEEE Network Magazine*, September, 1993.

AUTHOR'S BIOGRAPHY

Hani ElGebaly is a senior software architect with Intel Architecture Labs located in Oregon. He is the technical lead within the Broadband and IP Telephony lab for the voice over IP protocol development team. His current focus is on multimedia conferencing protocol development that complies with the International Telecommunication Union standards such as H.323.

Hani received a Ph.D. degree in computer science from the University of Victoria, Canada; an M.Sc. degree in computer science from the University of Saskatchewan, Canada; and a B.Sc. degree with high honors in electrical engineering from Cairo University, Egypt. Hani has contributed to multiple international standards and profiles for the telecommunication industry through the International Telecommunication Union (ITU) and other telecommunication standards' bodies.