Proc. Interspeech 2004 (Jeju Island, Korea), pp. 633-636, Oct. 2004

Synchronization of Speaker Selection for Centralized Tandem Free VoIP Conferencing

Colm Elliott *Peter Kabal*

Department of Electrical & Computer Engineering
McGill University, Montreal, Canada H3A 2A7

Abstract

Traditional teleconferencing uses a select-and-mix function at a centralized conferencing bridge. In VoIP environments, this mixing operation can lead to speech degradation when using high compression speech codecs due to tandem encodings and coding of multi-talker signals. A tandem-free architecture can eliminate tandem encodings and preserve speech quality. VoIP conference bridges must also consider the variable network delays experienced by different packetized voice streams. A synchronized speaker selection algorithm at the bridge can smooth out network delay variations and synchronize incoming voice streams. This provides a clean mapping of the N input packet streams to the M output streams representing selected speakers. This paper presents a synchronized speaker selection algorithm and evaluates its performance using a conference simulator. The synchronization process is shown to account for only a small part of the overall delay experienced by selected packets.

1 Introduction

Conferencing capability is an essential value-added service for any modern voice communication network. In conventional telephony, teleconferencing is implemented as a select-and-mix function at a centralized conference bridge. This same configuration is often transplanted to VoIP networks where the bridge has to consider the effect of variable network delays across packetized input voice streams.

In order to avoid tandem encodings and any coding of multi-speaker signals, a tandem-free arrangement can be used in which the bridge chooses M streams as speakers from N conferees, and forwards the M streams to endpoints without mixing them. This arrangements avoids the quality loss due to tandem coding (decoding and re-encoding a signal generally adds distortion) and due to the multiple talker mixed signal as input to a speech coder designed for single speakers. In a tandem-free arrangement are then responsible for decoding and mixing speech packets from the forwarded streams [1].

In traditional telephony, all speech is coded with PCM based waveform codecs and delay jitter experienced over the network is on the order of microseconds per sample. Transporting voice over IP networks introduces hybrid encodings of speech packets that experience network delay jitter on the order of packet lengths (20 ms or more). In a conferencing scenario, these delays and delay jitters may vary considerably from endpoint to endpoint.

Because of varying network delays, a synchronization mechanism is required at the bridge if it is to properly map the N input voice streams to M output streams representing currently selected speakers. Such a synchronization mechanism would also allow for the bundling of those M output stream into a single IP packet, so as to maintain a one-to-one connection model between the bridge and each endpoint [1].

This paper presents a synchronized speaker selection algorithm that selects M speakers from N input voice streams, while mapping N input Real Time Transport Protocol (RTP) streams to M output RTP streams, and maintaining a periodic and synchronized output across those M streams. A software simulator is used to evaluate the average end-to-end delay experienced by selected packets in a synchronized conferencing environment. The same simulator is also used to evaluate delay experienced by selected packets in a Select-and-Forward conferencing environment [1] in order to determine the additional delay required to perform synchronized speaker selection.

2 Tandem-Free VoIP Conferencing

Conventional centralized conferencing bridges perform a decode-mix-encode operation, resulting in a tandem encoding of a multi-speaker signal. Tandem encodings of high compression speech codecs such as G.729 and G.723.1 lead to additional speech degradations [2]. Tandem-Free Conferencing seeks to eliminate tandem encodings by offloading mixing duties to conference endpoints.

A tandem-free conferencing bridge will perform speaker selection on incoming packets by selecting M (usually 2) of N conferees as speakers at any one time. Packets for currently selected speakers are then forwarded to conference endpoints without decoding. Endpoints are responsible for decoding and mixing the M selected packet streams. In order for the bridge to be able to perform speaker selection *without* decoding packets, speech energy and a Voice Activity (VAD) decision must be sent as side information on upstream packets [1].

In addition to avoiding any degradations in speech quality due to tandem encodings and encodings of multi-speaker signals, the tandem-free architecture also eliminates the processing requirements of the decode-encode step found in conventional bridges, which can be large for high complexity codecs such as G.729 and G.723.1 [2].

Tandem-free operation is well suited to secure conferencing environments, as the speech payload can remain encrypted from source to destination. The bridge need not decrypt packet payloads in order to do speaker selection, as long as the side information is in cleartext (or encrypted with another key). This removes the requirement that the bridge be a “trusted” entity, as is the case in conventional centralized conferencing architectures.

The Select-and-Forward bridge, a tandem-free bridge suitable for use in centralized VoIP conferences is presented in [1]. The Select-and-Forward bridge performs speaker selection on individual packets as soon as they arrive, and immediately forwards selected packets to endpoints. Selected packets do not experience any buffering delay at the bridge. The Select-and-Forward bridge acts as a packet reflector, mapping N input streams to N output streams, M of which are active at any given time. While the Select-and-Forward design minimizes delay experienced by packets at the bridge, it creates an N-to-1 connection model between the bridge and endpoints.

3 Synchronized Speaker Selection

Synchronizing the speaker selection process across all input voice streams provides two main benefits over the Select-and-Forward design:

- A mechanism for mapping N input streams to M output streams, while ensuring a constant and periodic output across those M streams.
- The ability to bundle (or alternatively mix) the M output streams, creating a one-to-one connection model with endpoints.

Synchronization of incoming voice streams at the bridge allows the conference details, such as number of conference participants and delay characteristics between other endpoints of the bridge, to be abstracted from endpoints. This reduces endpoint complexity and provides added robustness in the face of sampling clock skew between endpoints.

3.1 Synchronization Algorithm

The synchronization process involves an intrastream synchronization stage followed by an interstream synchronization stage. The intrastream stage aims to remove network delay jitter incurred by a given stream on the source to bridge path, while the interstream synchronization will line up voice packet boundaries across all N input streams.

Intrastream Synchronization

Instrastream synchronization involves buffering an incoming voice stream so as to ensure that the input to the speaker selection process is continuous and periodic across a given voice stream. This process is analogous to playout scheduling done at VoIP endpoints. Each packet arriving at the bridge is assigned a *forwarding time*, which corresponds to the time it will be passed on to the speaker selection process, and forwarded to endpoints if selected as a speaker. Forwarding times of successive packets should be one packet period apart, so as to ensure a periodic output from the intrastream synchronization stage. In practice, some packets will not arrive in time for their scheduled forwarding time and are thus considered *late*. The tradeoff between buffering delay and the late rate of packets is parametrized by a target late rate l. The target forwarding time, t_i, for packet i is calculated as

$$t_i = a_i + (\tilde{D}_l - n_i), \qquad (1)$$

where a_i is the arrival time of packet i, n_i is the network delay incurred by packet i, and $\tilde{D}_l$ is an estimate of a delay value parameterized by target loss rate l such that

$$P[n_i < \tilde{D}_l] = 1 - l. \qquad (2)$$

$\tilde{D}_l$ can be estimated using traditional playout scheduling techniques such as those proposed in [3, 4, 5], and will be updated as packets arrive, so as to adapt to any changing network conditions.

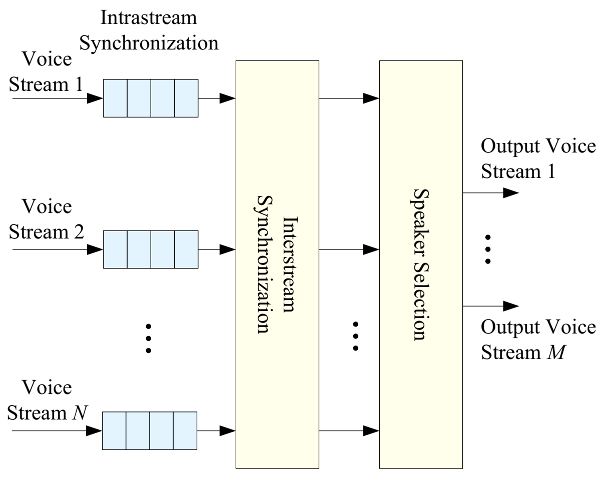

Fig. 1 Synchronized speaker selection.

For the first packet arriving in a stream, the forwarding time f_1 of that packet is the target forwarding time, t_1, as calculated in Eq. 1. Any subsequent packet is given a forwarding time based on the previous packet, in order to preserve periodicity:

$$f_i = f_{i-1} + T \pm mT, \qquad (3)$$

where T is the packet length, and $m = 0$ for any packet not classified as silence. Packets containing silence can be dropped or repeated if the forwarding times for a given voice stream need to be adjusted because of changing network conditions. During silence periods, m is chosen so as to minimize the difference between f_i and the target forwarding time t_i. If silence suppression is being used at endpoints (i.e., endpoints are not transmitting packets in times of silence), then the first packet that arrives after a silence period can be treated as if it were the first packet in the voice stream.

The choice of target late rate l is a tradeoff between the level of synchronization and additional buffering delay. Low target late rates will yield tight synchronization between incoming voice streams at the cost of higher buffering delay. Higher late rates will reduce buffering delay, but as synchronization loosens, conference endpoints may see network delay “jumps” during speaker transitions, because the source to bridge network delay characteristics are less well abstracted from conference endpoints. This can lead to playout scheduling delay errors and additional packet loss at destination endpoints.

Interstream Synchronization

Interstream synchronization ensures that selected voice streams will be output periodically from the bridge, even when there is a change in which streams are selected as speakers. This is done by lining up packet boundaries over all input streams. One input stream is selected as a *master* stream, and all other streams have their forwarding time adjusted so as to line up packet boundaries

$$s_i = f_i + ((m_j - f_i) \mod T) \tag{4}$$

where s_i is the synchronized forwarding time for packet i, and m_j is the forwarding time of any packet in the master stream [6].

After a given stream has been synchronized to the master stream, it can do any further intrastream synchronization based on its synchronized forwarding time

$$f_i = s_{i-1} + T \pm\ mT \tag{5}$$

where again $m = 0$ for packets not classified as silence. This ensures that all ensuing packets for the stream in question will already be synchronized with other input streams after the intrastream synchronization and that $f_i = s_i$ for $i > 1$.

3.2 Speaker Selection Algorithm

Since forwarding times will be synchronized across all input voice streams, speaker selection is done once every packet period. The speaker selection algorithm used is the Multi-Speaker/Interrupter(MS/I) algorithm presented in [7]. This algorithm selects talkers based on order of activity, the power signal envelope, $\hat{E}_i$, and a "barge-in" threshold B_{th}, which sets the bar for speaker interruptions.

The power signal envelope is updated for each speaker based on the signal power $\bar{E}_i$ of the current voice packet. It is assumed that this is computed at the source endpoint and carried as side information.

$$\hat{E}_{i+1} = \max(\bar{E}_i, \beta\hat{E}_i + (1-\beta)\bar{E}_i), \tag{6}$$

where β is the weight of the exponential average.

Speakers are ranked based on their power signal envelope, with the first M selected as speakers. To prevent spurious switching, a barge-in threshold, B_{th} is used such that a currently unselected voice stream's $\hat{E}_i$ must be greater than a selected voice stream's $\hat{E}_i$ by a factor of B_{th} in order to preempt that stream as a selected speaker.

3.3 Late Packets

If a packet from a given voice stream, j, arrives late for its scheduled forwarding time and is thus not available for speaker selection, no update on $\hat{E}_i^j$ is done, and the last available $\hat{E}_i^j$ is used for determining speaker rank. When the packet finally does arrive, $\hat{E}_i^j$ can be updated. If successive packets are late and have not yet arrived, $\hat{E}_i^j$ should be exponentially decayed so as to favour streams whose packets are not arriving late.

If the bridge is treating all M speakers separately (i.e., the bridge is not bundling selected streams in the same packet), then late packets for streams that were selected as speakers for the interval corresponding to that packet can be forwarded immediately upon arrival.

If the bridge is bundling selected speakers and forwarding them in one packet, a decision has to be made on whether to wait for selected streams that have late packets, or to generate an approximation of that voice streams contribution via some Packet Loss Concealment (PLC) mechanism and forward the bundled packet as per its scheduled forwarding time. If the late packet in question is selected as the *primary* speaker, then forwarding is deferred until the arrival of that packet. Otherwise, a PLC scheme can be used to approximate the late packet so as to avoid incurring any additional delay. Alternatively, a power signal envelope threshold can be used in determining whether it is worth waiting for a selected late packet or simply approximating that selected streams contribution. This approach can also be used if mixing selected packets instead of bundling.

4 Evaluation

4.1 Conference Simulator

The effects of the synchronization process on speech quality were evaluated using a conference simulator. Audio was taken from an actual four person conference and each conferee's contributions were isolated into four separate audio files. These served as inputs to four simulator endpoints. Connections between endpoints and the bridge were modelled with different network delay traces. Different bridge designs could then simulate the requisite speaker selection and packet forwarding, so as to give network delay and packet loss statistics across all voice streams as well as simulated conference output audio for each endpoint. Several different endpoint playout scheduling algorithms can be specified as parameters to the conference simulator.

4.2 Experimental Method

The baseline evaluation for average network delay over all selected packets for a given set of network traces was determined using a tandem-free Select-and-Forward bridge in the conference simulation.

The delay penalty incurred due to synchronized speaker selection is determined by simulating a conference using a Select-and-Forward bridge, and then simulating the same conference under the same network conditions with a synchronized speaker selection bridge. The average delays experienced by selected packets in each simulation can then be used to quantify the additional delay required to perform speaker synchronization.

Simulations were run over four sets of network traces, representing four different network environments for the conference. Some represented conferences where the network delay distributions were similar for all conference participants, while others represented more disparate network delays among conferees. Each simulation was run once with endpoints using a talkspurt-adaptive playout algorithm [3] and once with endpoints using a packet adaptive playout scheduling algorithm [5]. A packet size of 20 ms was used and 2 speakers were selected by the bridge per packet period ($M = 2$). Bridges were evaluated based on the end-to-end delay vs. packet loss characteristics averaged over all selected packets and all con-

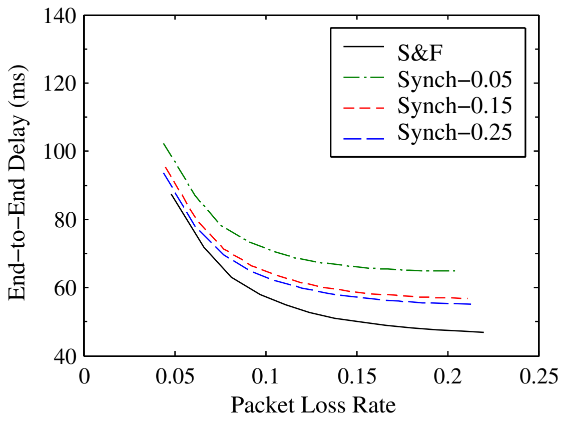

Fig. 2 Delay loss curves for several bridges with trace set 2 and packet adaptive playout algorithm at endpoints.

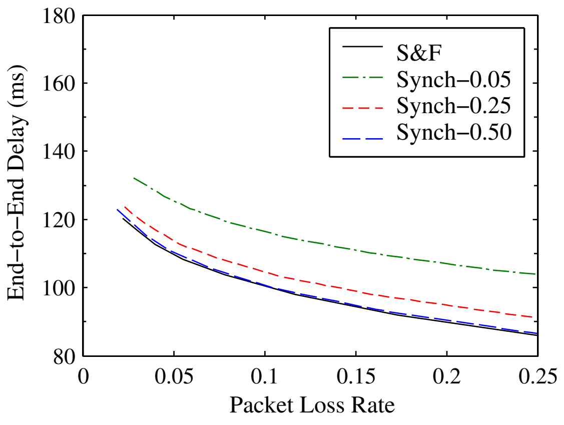

Fig. 3 Delay loss curves for several bridges with trace set 1 and talkspurt adaptive playout algorithm at endpoints.

ference endpoints. Synchronized bridges with target late rates of 5%, 15%, 25%, and 50% were evaluated, so as to examine the tradeoff between delay and tightness of synchronization.

4.3 Results

For all but one trace set, conferences using synchronized speaker selection with a target late rate of 50% experienced higher packet loss because endpoint to bridge network delay distributions differed enough from endpoint to endpoint to cause playout scheduling errors at endpoints after speaker transitions. Bridges using synchronized speaker selection with target late rates of 15% and 25% typically showed delay penalties on the order of half a packet length (10 ms) as compared to the performance of the Select-and-Forward bridge (see Fig. 2).

For trace set 1, shown in Fig. 3, synchronized speaker selection with a target late rate of 50% performs as well as the Select-and-Forward bridge in terms of average network delay. This is due to all endpoint to bridge paths having similar delay distributions and the endpoint playout algorithm's reliance on delay variance in calculating playout delay.

5 Conclusions

In general, synchronizing the speaker selection process can degrade speech quality of the resulting conference by injecting additional delay in the voice stream. The overall delay penalty is usually much smaller than the actual buffering delay experienced at the bridge, however. While adding delay, the synchronization process removes delay jitter from the source to bridge network path. This in turn lowers the playout delay at conferencing endpoints. The overall delay penalty is a function of the buffering delay incurred at the bridge, network delay characteristics and the playout scheduling algorithms used at conference endpoints. Conferences using endpoints that have playout scheduling algorithms that can adapt quickly to changing network conditions typically incur a greater penalty due to synchronization of speaker selection [5, 8]. Conferences using endpoint playout scheduling algorithms that explicitly use the measured network delay variance to estimate a playout deadline, such as [3], may actually perform better with a synchronized bridge, as the delay variance from the source to bridge path is removed by the synchronization mechanism.

References

[1] P. J. Smith, P. Kabal, M. Blostein, and R. Rabipour, "Tandem-free VoIP conferencing: A bridge to next generation networks," *IEEE Commun. Mag.*, vol. 41, pp. 136–145, May 2003.

[2] Cisco Systems, "Voice over IP overview." [online] `http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/1700/1751/1751swg/intro.pdf`.

[3] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne, "Adaptive playout mechanisms for packetized audio applications in wide area networks," *Proc. Conf. Computer Commun.* (Toronto, ON), pp. 680–688, June 1994.

[4] S. B. Moon, J. Kurose, and D. Towsley, "Packet audio playout delay adjustment: Performance bounds and algorithms," *Multimedia Systems*, vol. 5, pp. 17–28, Jan. 1998.

[5] Y. J. Liang, N. Färber, and B. Girod, "Adaptive playout scheduling using time-scale modification in packet voice communications," *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing* (Salt Lake, UT), pp. 1445–1448, May 2001.

[6] Y. Ishibashi and S. Tasaka, "A group synchronization mechanism for live media in multicast communications," *Proc. Conf. Computer Commun.* (Kobe, Japan), pp 692–700, April 1997.

[7] P. Smith, P. Kabal, and R. Rabipour, "Speaker selection for tandem-free operation VoIP conference bridges," *Proc. IEEE Workshop Speech Coding* (Tsukuba, Japan), pp. 120–122, Oct. 2002.

[8] A. Shallwani and P. Kabal, "An adaptive playout algorithm with delay spike detection for real-time VoIP," *IEEE Canadian Conf. Electrical, Computer Engineering* (Montreal, QC), pp. 997–1000, May 2003.