Voice-TFCC: a TCP-Friendly Congestion Control scheme for VoIP flows

Abdelbasset Trad
PRINCE Computer Science Research Unit
INFCOM Sousse, Tunisia
Email: trad.abdelbasset@gmail.com

Hossam Afifi
National Institute of Telecommunications
INT Paris, France
Email: hossam.afifi@int-evry.fr

Abstract—Typically, VoIP traffic is deployed as best-effort traffic over Internet links. This voice traffic lacks effective and scalable end-to-end congestion control. We propose a new VoIP congestion control scheme called Voice-TFCC (Voice TCP-Friendly Congestion Control), that tries to keep the transmission protocol overhead to a minimum while maintaining a TCP-friendly throughput. Voice-TFCC adjusts packet and codec rate in order to reduce the traffic load on Internet routers and the overall header bandwidth used by VoIP. We used analytical results to show the bandwidth efficiency obtained through our proposal. Voice-TFCC scheme is scalable because no changes are needed at core routers and minimal control messages are used and thus can be easily implemented and deployed in today's Internet.

I. INTRODUCTION

Given the continuous VoIP deployment, the Internet is expected to carry a significant proportion of the world's telephony traffic. Typically, voice traffic is deployed as best-effort traffic over Internet links. However, this best-effort voice traffic lacks effective and scalable end-to-end congestion control. These reasons led us to focus on the case of a large number of VoIP sources at an access network, sharing a common path over IP backbone networks and destined to different users in remote networks. We address the need to design congestion control for the growing class of VoIP traffic. This need is motivated by the inefficient use of network bandwidth caused by protocol header overhead of voice packets and the fairness problem caused by the transmission of large number of small VoIP packets sharing network links with TCP traffic. In fact, delay-sensitive applications such as VoIP typically use UDP as transport protocol, not implementing any type of congestion control. When VoIP applications share network resources with other applications that use a congestion control algorithm (i.e., TCP flows), a poor utilization of the bandwidth usually results leading, in the worst case, to a starvation of the network resources. The said problem has stimulated new ideas from both protocol and network designers, which follow two approaches. One approach is to implement an intelligent queue management at Internet routers considered as the point of congestion. The second approach is to implement an end-to-end congestion control specifically conceived for real-time applications. Several techniques have been proposed to understand the in depth nature of congestion control [1][2]. They rely on mathematical models to catch the relevant aspects of a general congestion control problem, but these techniques are usually very complex to implement. We propose a novel VoIP congestion control scheme called Voice-TFCC that combines RTP voice flow multiplexing, codec rate adaptation and the TCP-friendly congestion control mechanism. Voice-TFCC mechanism is applied on VoIP flows transmitted between two intermediate gateways. To the best of our knowledge our proposal is the first scheme that incorporates packet and codec rate adaptation for RTP flows while maintaining TCP-friendliness. We used analytical results to show the bandwidth efficiency obtained through our proposal. We believe that Voice-TFCC scheme combined with accurate estimation of network congestion state provides a significant step towards an efficient and scalable congestion control mechanism for VoIP traffic. The paper is organized as follows. Section II states the problem. Following, in Section III we discuss related work. Section IV presents the Voice-TFCC scheme. In Section V, we analyze the dynamics of our proposal. Section VI illustrates Voice-TFCC performance. Finally, Section VII concludes the paper.

II. PROBLEM STATEMENT

In order to reduce bandwidth usage, low-bit-rate voice codecs are used in IP telephony systems. The commonly used are G.711, G.726, G.729A and G.723.1 (Table I). Voice frames are transmitted through the IP network based on UDP and the Real-time Transport Protocol (RTP/RTCP) [4][5]. Although RTP does not contain any mechanism that guarantees the timely delivery of data, RTCP reports provide a regular summary statistics in order for the sender and the receiver to adapt their behavior to the current network congestion state. Since voice packets are encapsulated with IP/UDP/RTP headers it can be assumed that, for a packet duration of 20 ms, header information will add 16 kbps to the bandwidth requirement for VoIP flow (Table I). The voice codec defines the size of the sample but the total number of samples placed in a packet affects the rate of packets sent per second. This number is another factor in determining the bandwidth of a voice call. A number of samples representing 30 ms is considered to be the maximum duration for the payload. This duration is a compromise between bandwidth requirements and quality. In fact, smaller payloads demand higher bandwidth per channel band and decrease transmission efficiency. On the other hand,
when the payload size increases, the overall latency of each call also increases since voice samples have to be buffered for a longer period of time. When payloads are increased the system will also be more susceptible to the loss of individual packets by the network. Although, available bandwidth in current IP-based backbones is often sufficient enough for a limited number of VoIP communications, a large number of voice communications transmitted between two edge gateways of an IP backbone network will cause inefficient use of network resources due to packet header overhead and bursts of small voice packets. Current VoIP applications tackle this problem by embedding multiple audio frames into a single packet at the source to increase the ratio of payload to header size. This approach has the benefit of reducing the overall data rate of a call. But, packing an additional audio frame will add another frame period to the assembly delay. Together with the existing network delay, resultant end-to-end delays may become unacceptable [3]. In local area networks where bandwidth is abundant, VoIP applications can send each audio frame in a separate RTP packet to minimize packetization delay. However, in case of Internet telephony gateways with multiple RTP streams, the bandwidth that the header occupies must be taken into consideration. From another point of view, UDP traffic is unresponsive to congestion and therefore can completely monopolize the available bandwidth. Thus, Internet load increases because of large numbers of short voice UDP packets, with 100 packets flowing every second in both directions for each call into the IP network, eventually resulting in large delay, jitter and packet loss. Performance problems will be experienced, in this case, by all voice calls and also by other traffic (i.e., TCP traffic) sharing the best-effort IP network.

### III. DISCUSSION OF RELATED WORK

In this section, we discuss RTP multiplexing and TCP-friendly congestion control mechanisms that we have coupled to develop Voice-TFCC scheme.

#### A. RTP Multiplexing Schemes

Multiplexing aims at the reduction of the overhead associated with Internet protocol layers and traffic load (i.e., number of packets) on routers. The basic assumption for the multiplexing is that, at any time, there is more than one user communicating with the same remote location. Various approaches for multiplexing VoIP streams between peer gateways have been proposed [3][6].

The total number of voice-packets that are multiplexed should be obtained by considering trade-offs between delay and bandwidth efficiency improvements. Indeed, the main drawback of these multiplexing schemes is the added delay. Two types of delay are incurred. Multiplexing processing delay occurs in the gateway given that the generation of a multiplexed packet is triggered by the expiration of the multiplexing period or by the arrivals of enough voice packets to fit into the MTU. The multiplexed packet will also encounter additional packet transmission delay at Internet routers since most network limitations in today’s Internet are in bytes per second. With regard to packet loss, multiplexing is likely to reduce losses on the Internet. First, because improved efficiency results in a reduced overall bit rate for the voice stream and thus the necessary bandwidth. Secondly, many routers drop packets not because of link congestion and buffer overflow, but because of processing overload. A burst of small packets can overwhelm the processors on a typical router, causing losses.

#### B. TCP-Friendly Rate Control for Unresponsive Flows

TCP-friendly equation-based rate control (TFRC) was introduced to ensure proper congestion avoidance for multimedia applications using unresponsive transport protocols. The Equation 1 developed in [7], roughly describes TCP’s sending rate as a function of the loss event rate, round-trip time and packet size. The loss event rate pis given by the number of packet loss events as a fraction of the number of packets transmitted.

\[
T(Bps) = \frac{S}{RTT \sqrt{\frac{24}{3}} + RTO (3 \sqrt{\frac{24}{3}}) p(1 + 32p^2)}
\]  

#### IV. PROPOSED VOIP CONGESTION CONTROL SCHEME

In a packet-based bottleneck bandwidth (each packet requires a single queue buffer regardless of the packet size), the decision to drop a packet is independent of the packet size. VoIP flows will then see similar packet drop rates as TCP flows, though VoIP paquets are smaller than TCP ones. Fairness results can change significantly if the drop-tail queue at the bottleneck link is in units of bytes rather than packets. In fact, a byte-based queue has a fixed number of bytes, and an almost-full queue might have room for a small packet but not for a large one [9]. Hence, large packets are more likely to be dropped than are small ones and VoIP flows will see a much smaller drop rate than TCP flows, and consequently receive a much larger sending rate. As stated in RFC 3714 [10], the ideal would be to have a transport protocol that is able to detect whether the bottleneck links along the network path are limited in bytes/sec or in packets/sec, and to respond appropriately, but such an ideal is hard to achieve. Thus, the deployment of

<table>
<thead>
<tr>
<th>Codec</th>
<th>Compression method</th>
<th>Bit rate (kbps)</th>
<th>Frame size (ms)</th>
<th>Payload size (bytes)</th>
<th>IP bandwidth (kbps)</th>
<th>IP/UDP/RTP Header overhead (%)</th>
<th>Equipment Impairment I_e</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.711</td>
<td>PCM</td>
<td>64</td>
<td>20</td>
<td>160</td>
<td>80</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>G.726</td>
<td>ADPCM</td>
<td>24/32/40</td>
<td>20</td>
<td>60/80/100</td>
<td>40/48/56</td>
<td>40/33.33/28.57</td>
<td>25/12</td>
</tr>
<tr>
<td>G.729A</td>
<td>CS-ACELP</td>
<td>8</td>
<td>20</td>
<td>20</td>
<td>24</td>
<td>66.66</td>
<td>11</td>
</tr>
</tbody>
</table>

**TABLE I**

**FEATURES OF MAIN STANDARD VOIP CODECS**
congestion control for telephony traffic should not be delayed until such an ideal could be accomplished. In this work, we design a new generic scheme named Voice-TFCC (Voice TCP-Friendly Congestion Control) that incorporates techniques for controlling both packet rate and codec rate of VoIP flows based on TCP-friendly congestion control mechanism. This protocol is applied to many VoIP flows transmitted between two VoIP gateways but can also be used by a single VoIP flow. Voice-TFCC adapts the control traffic (i.e., packet headers) to the network congestion state using TCP-friendly equation-based rate control mechanism. It uses a simple multiplexing scheme where header overhead is reduced through multiplexing several RTP streams destined to the same gateway into one UDP packet. The multiplex is formed by linking a series of RTP streams and an IP-UDP header. At fixed time intervals, the receiver computes the loss rate observed during the previous interval. The sender, based on the receiver’s feedback information, updates its sending rate by adjusting the number of packets to multiplex hence the packet rate. The number of RTP flows to multiplex is limited by MTU packet size.

A. Voice-TFCC Algorithm Description

The main idea behind our proposal is to decide the number of VoIP flows to be multiplexed and the codec rate to be used based on TCP-friendly decision. The goal of this decision is to reduce the packet rate and hence to reduce the traffic load on Internet routers. Consequently, the header bandwidth used by VoIP flows will also be reduced. When the maximum number of flows that can be multiplexed is reached, a second phase of the protocol is executed. This phase consists in transcoding VoIP flows in order to adapt the generated codec bit rate to the actual network bandwidth given by the TCP-friendly throughput estimation. Let \( n \) denotes the number of incoming VoIP flows at the sending gateway. Initially the number of packets to multiplex, \( m_0 \), is set to \( l \) and the packet sending rate is determined by the total rate of voice flows coming at the sending gateway. When a feedback message is received, the sender changes the number of packets to multiplex based on the information obtained from the receiver gateway. Assuming that at a transmission time period \( i \) the gateway is sending a flow of \( m_i \) multiplexed voice packets using codec \( i \) and a total throughput rate of \( T_{\text{ Codec}} \) bytes/sec, then after receiving the feedback message RTCP, from the receiver gateway the sender measures the round-trip time estimate, updates the retransmission timeout value. The loss rate obtained from the receiver gateway, \( p_i \), and the measured round-trip time, \( RTT_i \), are then fed into the throughput Equation 1, to give the new acceptable TFRC sending rate \( T_{\text{ TFRC}}^{i+1} \) \( \left( T_{\text{ TFRC}}^{i+1} = T(S(m_i), RTT_i, p_i) \right) \), where \( S(m_i) \) is the size in bytes of a multiplexed packet formed by \( m_i \) RTP voice packets and one IP-UDP header, given by the Formula 2:

\[
S(m_i) = h_{\text{ip}} + h_{\text{udp}} + m_i(h_{\text{rtp}} + \text{Payload}) \tag{2}
\]

For a given number \( n_i \) of incoming VoIP flows to be transmitted between the sender and receiver gateway, the total throughput generated when \( m_i \) flows are multiplexed with one header is given by Equation 3:

\[
B_{\text{mux}}(i) = m_i T_{\text{ codec}}^i + \left( \frac{n_i}{m_i} \right) T_{\text{ header}} \tag{3}
\]

where:

- \( T_{\text{ codec}}^i \) denotes the throughput in kbps, generated by the voice codec used (considering RTP header size and voice payload).
- \( T_{\text{ header}} \) denotes the throughput generated by IP-UDP header (i.e., 11.2 kbps)\(^1\).
- \( m_i \) denotes the number of RTP flows being multiplexed.

The basic idea behind the proposed mechanism is that the sender gateway should adjust the number of multiplexed RTP packets to have the throughput that matches the calculated rate \( T_{\text{ TFRC}}^{i+1} \) in the next transmission time period \( i, i+1 \). We propose to use Equation 4 to determine the number of packets to multiplex according to the network congestion state:

\[
B_{\text{mux}}(i+1) = T_{\text{ TFRC}}^{i+1} \tag{4}
\]

where \( B_{\text{mux}}(i+1) = n_{i+1} T_{\text{ codec}}^{i+1} + \left( \frac{n_i}{m_i} \right) T_{\text{ header}} \). The network congestion state is reflected by the TFRC throughput estimation \( T_{\text{ TFRC}}^{i+1} \) obtained after the transmission time period \( i, i+1 \). Basically, Voice-TFCC algorithm operates in two phases. In the Phase I, the codec rate is not changed between two successive time intervals. The number of flows to multiplex \( m_{i+1} \) is then determined according to the Equation 5:

\[
m_{i+1} = \frac{n_{i+1} \times T_{\text{ header}}}{T_{\text{ TFRC}}^{i+1} - n_{i+1} T_{\text{ codec}}^{i+1}} \tag{5}
\]

The sender will increase the number of multiplexed RTP packets, \( m_{i+1} \), if there was a high traffic load during the previous time interval indicated by a calculated TFRC sending rate \( T_{\text{ TFRC}}^{i+1} \) less than the previous sending rate \( T_{\text{ TFRC}}^i \). Otherwise, the sender will decrease the number of multiplexed packets during normal load periods. If the resulting number of flows to be multiplexed is greater than the number of incoming flows \( m_{i+1} > n_{i+1} \) then all the incoming flows will be multiplexed \( m_{i+1} = n_{i+1} \) and the second phase of the Voice-TFCC algorithm is executed. The Phase II, consists in changing the codec rate in addition to packet rate. The codec rate to be used \( T_{\text{ codec}}^{i+1} \) will be determined using the expression given by Equation 3 for \( (m_{i+1} = n_{i+1}) \) while considering codec rate as a variable:

\[
B_{\text{codec}}(i+1) = T_{\text{ TFRC}}^{i+1} \tag{6}
\]

where \( B_{\text{codec}}(i+1) = n_{i+1} T_{\text{ codec}}^{i+1} + T_{\text{ header}} \). The new codec bit rate will be chosen among coding rates available at the sender gateway to best conform Equation 7:

\[
T_{\text{ codec}}^{i+1} = \frac{T_{\text{ TFRC}}^{i+1} - T_{\text{ header}}}{n_{i+1}} \tag{7}
\]

\(^1\)IPv4 header of 20 bytes and UDP header of 8 bytes, generated each 20 ms of voice frame period.
The second phase of the algorithm is also executed if the resulting number of flows to be multiplexed is less than one flow ($n_{i+1} < 1$). This case indicates that the network bandwidth is sufficient for the transmission of $n_i$ VoIP flows without the need for multiplexing and that the codec rate may be increased according to Equation 4 ($m_{i+1}$ is set to 1) during Phase II of the algorithm. The Phase II is executed in another case where the packet formed by multiplexing $m_{i+1}$ flows using the codec throughput $T_{i+1}^{\text{codec}}$ will exceed the MTU size. In that case, the $m_{i+1}$ is set to the maximum possible number of flows to multiplex and codec bit rate is reduced using Equation 6. Voice-TFCC scheme provides a generic TCP-friendly congestion control mechanism that can be used in the case of a single VoIP flow. The codec rate can be adapted using Equation 6. Figure 1 represents the flow chart of the VoIP sources. Actually, Voice-TFCC sends out a multiplexed packet every $T_m$(ms), which is equal to or shorter than the VoIP inter-packet interval (typically 20 ms). Larger values of $T_m$ can improve bandwidth efficiency since more packets can be multiplexed, but the delay incurred will also be larger. For example, if $T_m = 10$ ms, every two multiplexed packet contains one voice packet from each VoIP stream. The maximum introduced multiplexing time for one voice packet is 10 ms. If $T_m = 20$ ms, every multiplexed packet contains one voice packet from each VoIP stream, and the maximum introduced multiplexing time is 20 ms. Adjusting the multiplexing time $T_m$, is another factor that can be used to control the tradeoff between bandwidth efficiency and delay. In this work, we will not study the dynamics related to this factor. We consider that this time is constant. Although, packet loss rate may increase for large packet sizes especially in byte-based network environments, robustness against packet losses will not be affected given that for a given VoIP source, RTP packet $(i)$ and RTP packet $(i+1)$ are transmitted in separate multiplexed IP packets.

**B. Discussion of the Variable Size of Voice-TFCC Packets**

Originally, TCP-friendly rate control mechanism was designed for applications that use fixed packet size, and vary their sending rate in packets per second in response to congestion. TCP-friendly rate control mechanism should not be used for applications that vary their packet size instead of their packet rate in response to congestion [8]. Varying the packet size during the time interval between two estimations of the sending rate distorts packet-based measurement of the loss event. Voice-TFCC adapts its sending rate by adjusting the number of multiplexed packets, consequently the packet size is varied. However, Voice-TFCC varies the packet size only after the estimation of the sending rate using the TCP-friendly throughput equation and keeps this size fixed until the next feedback message. Therefore, Voice-TFCC sending rate estimation is quite accurate.

**VI. EVALUATION OF VOICE-TFCC SCHEME**

In this section, we illustrate through analytical results the performance of our proposal.

**A. Saving Bandwidth by Voice-TFCC Scheme**

In a bandwidth limited network environment, the resource utilization will depend on the packet size. Without multiplexing, the bandwidth required for the transmission of $n$ RTP voice packets is given by:

$$B_n = n \times (h_{ip} + h_{udp} + h_{rtp} + \text{Payload})$$

(8)

With Voice-TFCC scheme, the bandwidth required for the transmission of the same amount of voice data by multiplexing $m$ RTP voice packets into one UDP packet (assuming that $n$ is a multiple of $m$) is given by:

$$B_{n,m} = \frac{n}{m} \times (h_{ip} + h_{udp} + m \times (h_{rtp} + \text{Payload}))$$

(9)

---

**Fig. 1. Flow chart of the Voice-TFCC scheme**

Voice-TFCC framework parameters could be tuned according to the underlying network technology. For example, in the case of limited bandwidth on a wired Internet link, both coding and packet rate have to be adapted to have an optimal quality. However, on a link technology with large packet switching overhead such as in IEEE 802.11 wireless LANs, the packet rate should be lowered but the coding rate can not be reduced.

**V. VOICE-TFCC SCHEME ANALYSIS**

In this section, we discuss issues related to the proposed Voice-TFCC scheme.

**A. Delay and Loss Rate of Voice-TFCC Packets**

Since the multiplexed Voice-TFCC packet is formed by RTP frames originating from different sources and flowing at a given time at the sender gateway, packet delay will be increased just by a small multiplexing interval (< 20 ms) required for synchronization purposes between the different
The bandwidth saved by multiplexing $m$ RTP voice packets can then be calculated from Equation 10:

$$\beta_m = \frac{B_n - B_{n,m}}{B_n} = \frac{1}{m} \frac{(h_{ip} + h_{udp})}{h_{ip} + h_{udp} + h_{rtp} + \text{Payload}}$$

Figure 2 plots the percentage of bandwidth saving vs. the number of multiplexed packets for a typical payload sizes of 20 bytes and 160 bytes. We notice that the bandwidth saved increases quite significantly when the number of multiplexed packets is varied from 1 to 10 packets. With 10 multiplexed packets having a payload of 20 bytes, a significant bandwidth saving of 42% is achieved. Note that the multiplexed-packet length is bounded by the MTU. In Figure 3, we show the effect of payload size on the bandwidth saving obtained by multiplexing. It is important to notice that multiplexing smaller packets allows more bandwidth saving, for a given number of multiplexed packets. In the second phase of Voice-TFCC algorithm, the codec rate is changed but the number of multiplexed packets is kept the same, thus the bandwidth gain is obtained from payload size reduction and not header overhead reduction. The Equation 1 used for the TCP-friendly rate estimation is proportional to $\frac{S}{R_T T \sqrt{p}}$ where $S$ is the packet size and hence allows more sending rate when bigger packets are transmitted. However, in bandwidth-limited network environments, this behavior will be penalized by packet delay increase due to queuing delays occurring especially in case of network congestion. The delay increase will result in a reduction of the TCP-friendly rate estimation and hence in a reduction of the number of VoIP flows that can be accepted for the transmission between VoIP gateways. For that reason, codec rate adaptation is important. It allows the transmission of large number of VoIP flows between the VoIP gateways with a slight quality degradation due to the equipment impairment related to low bit rate codecs.

VII. Conclusion

In this paper, we addressed the need to design congestion control for the growing class of VoIP traffic. We designed a novel scheme called Voice-TFCC for joint control of VoIP coding and packet rate. The packet rate is dynamically adjusted through multiplexing and the codec bit rate is adapted using different audio codecs based on TCP-friendly rate control mechanism. Thus, in both bandwidth and packet rate bottleneck environments, Voice-TFCC enhances network utilization. Voice-TFCC scheme uses TCP-friendly rate control mechanism in order to ensure fairness with current Internet traffic. Voice-TFCC achieves performance goals of voice flows as well as efficient network utilization and fairness with TCP traffic. Our scheme is scalable because no changes are needed at core routers and minimal control messages are used and thus can be easily implemented and deployed in today’s Internet.

REFERENCES