Proceedings of the 7th IFIP Workshop on Performance Modelling and Evaluation of ATM/IP Networks, Antwerp, Belgium, 28-30 June 1999.

# Voice over IP in Access Networks

D. De Vleeschauwer, J. Janssen, G.H. Petit
Alcatel Bell, Corporate Research Center
Francis Wellesplein 1
B-2018 Antwerp, Belgium
Tel: ++32-3-240.81.96 - Fax: ++ 32-3-240.99.32
E-mail: {danny.de_vleeschauwer, jan.janssen, guido.gp1.petit}@alcatel.be

**Abstract**

In the context of the integration of voice, video and data traffic on one backbone, transporting voice calls over an IP network has recently caught a lot of attention. More precisely, it remains to be determined which techniques are required to guarantee traditional quality to voice calls. Depending on the way the voice signal accesses the IP backbone several voice communication scenarios can be distinguished. In this paper we concentrate on the PC-to-PC scenario. In this scenario voice calls are routed to the IP backbone over IP-based access networks, whose limited resources have to be shared between the voice call and the simultaneously active data applications. We study the impact of the choice of the codec, the bit rates of the access network, the characteristics of the data applications simultaneously active with the voice call and the access transport protocol on the effective bit rate needed to conduct a call of traditional quality. Even if voice packets are given head-of-line priority over data packets, guaranteeing traditional quality to voice calls becomes problematic with traditional IP-over-PPP if the access bit rates are too low. We show that to alleviate this problem access transport protocols as IP-over-fragmentation-oriented-PPP, IP-over-suspend/resume-oriented-PPP or IP-over-ATM are required. We also consider (IP/UDP/RTP) header compression and conclude that it is solely useful to reduce the effective bit rate required for the transport of the voice call, but brings no solution if guaranteeing traditional quality is problematic.

## 1. Introduction

Routing voice calls over an IP* network is at the centre of attention recently [5, 7, 9]. The ultimate goal is to integrate the traffic generated by voice, video and data sources on a common IP network rather than maintaining a separate network for those kinds of traffic. Because the quality of any real-time service heavily depends on the delay incurred from source to destination and traditional IP networks are notorious for the delay they introduce, guaranteeing QoS* to voice calls over IP networks remains an open issue. For future IP networks to be able to guarantee QoS to real-time services, they will have to be managed and will have to give real-time packets some kind of priority over non-real-time packets. In this paper we assume that voice packets get HOL* priority over data packets and that the number of voice calls on the IP network is limited in some way. Even in such a managed IP network with priorities several issues remain open. First, the voice payload size has to be chosen: large payloads may jeopardise the timely delivery of the voice signal, small payloads may exploit the network resources too inefficiently. Second, because some voice packets may be delayed more than others in the IP network, i.e. the flow is jittered, and because the voice decoder needs a constant payload stream, a dejittering mechanism is necessary. The dimensioning of this dejittering mechanism is a trade-off between payload loss and delay: the more the play-out time of payloads is postponed, the fewer payloads are lost due to late arrival, but the larger the mouth-to-ear delay gets. Finally, since the mouth-to-ear delay a user is willing to tolerate heavily depends on the level of the disturbing echo, the amount of echo control is also a remaining issue.

Depending on the way the voice information is transported to the edge of the IP backbone several VoIP* communication scenarios can be distinguished. In the phone-to-phone scenario the VoIP call originates and terminates in an ordinary telephone. The voice signal is switched over the traditional PSTN* to an ingress gateway at the edge of the IP backbone. In this ingress gateway the voice signal is encoded and packetised. The resulting packet flow is then routed over the IP backbone to an egress gateway. In this egress gateway the packet flow is dejittered and decoded. Finally, the voice signal is switched over the traditional PSTN from the egress gateway to its destination phone. Echo control is performed in both gateways to compensate for echo, particularly the electrical echo introduced in the 4-to-2-wire hybrids in the PSTN. The performance of some commercially available gateways is compared in [7]. In the PC-to-PC scenario the call originates and terminates in a PC. The functionality of the gateways of the phone-to-phone scenario now resides in the PCs. The crucial difference with the phone-to-phone scenario is that the PCs have access to the IP backbone through a relatively slow access line that needs to be shared between the VoIP call and all simultaneously active data applications (e.g. web browsing, exchanging files with the other caller, …). In this paper, we assume that there is only one voice call per access line. Yet, the interference of the data packets may be so large that special access transport protocols may be required in the access network. A third scenario, which only received little attention in the recent literature, is the mobile-terminal-to-mobile-terminal scenario in which the voice information accesses the IP backbone over wireless access networks. Further complications in this scenario are the following. First, transporting (voice) information over an air interface introduces a large delay. Second, the voice information is already encoded before it is transmitted over the air interface (e.g. GSM*) and possible transcoding at the edge of the IP backbone can have a detrimental effect on the voice quality. Finally, all mixed scenarios (phone-to-PC, PC-to-mobile-terminal, and mobile-terminal-to-phone) are also of importance.

In this paper we focus on the PC-to-PC scenario. More precisely, we determine the effective bit rate a VoIP call of traditional quality requires on an access network. We study the influence of the choice of the codec, the bit rates of the access network, the access transport protocol and the characteristics of the data application simultaneously active with the VoIP call.

In Section 2 the communication scenario under consideration is explained in detail. Section 3 deals with the mouth-to-ear delay that a user is willing to tolerate. Furthermore, it describes the delay components that are contributing to the mouth-to-ear delay. In Section 4 the payload size for a voice source is determined and the corresponding effective bit rate is calculated. Section 5 discusses some representative examples. In Section 6 some conclusions are drawn.

## 2. The voice over IP communication scenario

### 2.1. General

Figure 1 shows the VoIP communication scenario under study in this paper. In the originating PC the voice signal is encoded and the code words produced by the codec are packed in IP packets. These IP packets are transported over the IP network from the originating PC to the destination PC. The IP network consists of low-speed access networks and a high-speed backbone network. In the destination PC the voice flow is dejittered and decoded. Both PCs perform echo control, which is necessary to suppress the acoustic echo generated by the unwanted coupling of the loudspeaker signal into the microphone at the originating and destination PC. The acoustic echo can be large in this scenario since multimedia PCs are normally not optimised to have a large attenuation on the acoustic path from loudspeaker to microphone.

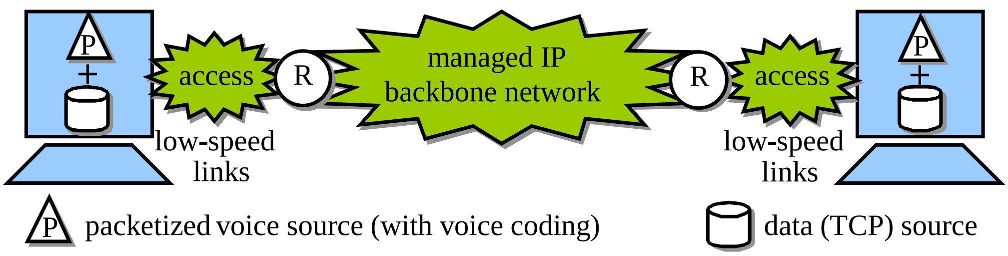



**Figure 1: The considered voice over IP scenario.**

We consider three codecs in this paper: the traditional G.711 codec, and the two low bit rate codecs usually considered in VoIP applications, the G.729 codec and G.723.1 codec. The latter codec can operate at two bit rates. These codecs are frame-based, i.e. they process speech intervals of $T_F$ ms, also referred to as voice frames, in one go. Every $T_F$ the encoder produces a code word of $B_F$ bits, which results in a codec bit rate of

$$R_{cod} = \frac{B_F}{T_F} \quad . \tag{1}$$

Table 1 gives the relevant parameters for the considered codecs.

Concerning the backbone network we assume

- that it is of very high speed,
- that voice packets get non-pre-emptive HOL priorities over data packets, i.e. an arriving voice packet leapfrogs over all queueing data packets, but has to wait for the data packet in service to finish, and
- that it is managed, i.e. the load on the network generated by all voice calls is limited by some call acceptance control mechanism.

Under these assumptions the time a voice packet spends in the backbone network is negligible with respect to the total mouth-to-ear delay.

### 2.2. The access networks

Two kinds of access networks are considered: a N-ISDN*-like network in which the up- and downstream bit rate are given by $R_{up}$=$R_{down}$=64 kb/s and an ADSL*-like network in which the up- and downstream bit rate differ. Both a low-speed and a high-speed ADSL network are considered. In the former case the up- and downstream bit rate are given by $R_{up}$=64 kb/s and $R_{down}$=512 kb/s, in the latter case the up- and downstream bit rate are given by $R_{up}$=256 kb/s and $R_{down}$=2.048 Mb/s.

We assume that during the VoIP call the user shares the available (up- and downstream) bit rate between his VoIP call and his data applications. We consider two types of data applications simultaneously active with the VoIP call: a file transferring application and web browsing. In the former case the interfering IP data packets in the up- and downstream direction are of the same maximum size, e.g. 576 bytes. In the latter case only acknowledgements of 40 bytes are sent in the upstream direction, while IP data packets of the maximum size (e.g. again 576 bytes) are downloaded in the downstream direction. It is clear that web browsing will have less influence on the VoIP call than a file transferring application, particularly in the case of an asymmetric, ADSL-like access network.

We consider two kinds of transport protocols to transport the IP packets over the access network: IP-over-ATM* [5] and IP-over-PPP* [8]. In the case of IP-over-ATM the IP packets (beit voice or data) are first encapsulated in AAL5* frames before they are segmented in ATM cells. Moreover, on the ATM layer voice cells are given non-pre-emptive HOL priority over data cells, i.e. an arriving voice cell leapfrogs over all queueing data cells, but may have to wait for a data cell in service. Remark that to be able to give this priority, it has to be possible to distinguish between voice and data cells. This means that voice and data will have to use a separate VC* (see the VC-mux alternative in [5]) and that the LLC* encapsulation alternative [5] is not adequate. Hence, there is no need for a LLC header in the AAL5 frame. In the case of IP-over-PPP, voice packets also have HOL priority over data packets. However, this may not be sufficient for the voice signal to be timely delivered, because a large data packet in service may still occupy the server for quite a long time. Therefore, we also consider different extensions of PPP: Fragmentation-oriented PPP (F-PPP*) [1] and Suspend/Resume-oriented PPP (S/R-PPP*) [2]. In F-PPP IP data packets are segmented in chunks of e.g. 200 bytes before they are transmitted over the access network. An arriving voice packet merely has to wait for a chunk of data of this size to terminate. In S/R-PPP an IP data packet in service is interrupted as soon as a voice packet arrives. Signalling the receiving side that the data packet was interrupted takes some time. We assume that an arriving voice packet effectively sees only 3 bytes of the interfering data packet.

Independently of the access transport protocol, IP-over-ATM or IP-over-PPP, header compression can be used. In this paper, we only consider the (IP/UDP*/RTP*) header compression, described in [3], for PPP-like transport protocols. In the latter contribution it is shown that the header of 40 bytes (20 bytes for IP, 8 bytes for UDP and 12 bytes for RTP) for real-time services can be compressed to 2 bytes on a point-to-point link.

## 3. Mouth-to-ear delay

### 3.1. Tolerable mouth-to-ear delay

If the packet loss is not too large, the quality of a phone call is mainly determined by the mouth-to-ear delay, the codec used and the level of the disturbing echo. In [4] the ETSI E-model was used to determine the quality of a phone call as a function of the (one-way) mouth-to-ear delay for various codecs and levels of disturbing echo. Although echo can have a detrimental effect on the quality, we consider the case of perfect echo control in this paper. Perfect echo control is possible (at moderate computational cost) if a combination of an echo canceller and a non-linear process is used.

| Standard | Frame size $T_F$ (ms) | Look-ahead $T_{LA}$ (ms) | Code word length $B_F$ (bits) | Bit rate $R_{cod}$ (kb/s) | Tolerable mouth-to-ear delay $T_{M2E}$ (ms) |
|---|---|---|---|---|---|
| ITU-T G.711 | 0.125 | 0 | 8 | 64 | 379 |
| ITU-T G.729(A) | 10 | 5 | 80 | 8 | 278 |
| ITU-T G.723.1 | 30 | 7.5 | 158<br>189 | 5.3<br>6.3 | 203<br>237 |

**Table 1: Relevant parameters for the considered codecs.**

From [4] we conclude the following in the case of perfect echo control.

- The quality of a phone call remains constant if the mouth-to-ear delay stays below 150 ms. This value of 150 ms is codec independent. Up to 150 ms the quality is solely determined by the intrinsic quality of the codec. For the codecs considered in this paper the G.711 codec has the best intrinsic quality, followed by the G.729 codec, the G.723.1 codec at 6.3 kb/s and the G.723.1 codec at 5.3 kb/s. All considered codecs attain traditional quality below 150 ms.
- Due to the loss of interactivity, the quality drops monotonically with the increasing mouth-to-ear delay for a mouth-to-ear delay above 150 ms. The mouth-to-ear delay for which the quality drops below traditional quality is codec dependent. These values, referred to as the tolerable mouth-to-ear delay, are given in Table 1 for the considered codecs.

### 3.2. Mouth-to-ear delay components

In the scenario of Figure 1 the mouth-to-ear delay is composed of the codec delay $T_{cod}$, the packetisation delay $T_{pack}$, the service delay $T_{serv}$, the queueing delay $T_{que}$, the dejittering delay $T_{jit}$ and the propagation delay $T_{prop}$.

For frame-based codecs, as all codecs in this paper, the codec and packetisation delays are closely intertwined. Frame-based codecs chop the speech signal in intervals of length $T_F$, referred to as voice frames, which they process one after the other. Collecting a voice frame introduces a delay of $T_F$. In order to make the coding process more efficient some encoders collect a further part of the voice signal, referred to as the look-ahead $T_{LA}$, after they have collected a voice frame. This introduces an additional delay of $T_{LA}$. Then, after an encoding time $T_{enc}$ the encoder produces a code word of $B_F$ bits. It may be inefficient to put only one code word in one IP packet. However, waiting for the encoder to produce more code words introduces additional delay: an additional $T_F$ per extra code word. At the receiver side the decoder needs a decoding time $T_{dec}$ to produce a close copy of the original voice frame from the code word it extracts from an IP packet. Hence, codec and packetisation consume a delay given by

$$T_{enc} + T_{LA} + N_F T_F + T_{dec} \quad , \tag{2}$$

in which $N_F$ is the number of code words put in one IP packet.

We define the codec delay as the sum of the look-ahead $T_{LA}$, the encoding delay $T_{enc}$ in the originating PC and the decoding delay $T_{dec}$ in the destination PC. Because the codec has to work in real-time, both the encoding $T_{enc}$ and decoding delay $T_{dec}$ are upper bounded by the voice frame length $T_F$. We assume that the processors in the PCs give priority to the encoding and decoding process and are so fast that ample processing power remains for the processes simultaneously active with the VoIP call. Hence, we assume that the codec delay $T_{cod}$ can be approximated as follows,

$$T_{cod} = T_{enc} + T_{dec} + T_{LA} \approx T_{LA} \quad . \tag{3}$$

With this definition of the codec delay, the packetisation delay is proportional to the number $N_F$ of code words packed into one IP packet, i.e.

$$T_{pack} = N_F T_F \quad . \tag{4}$$

The voice payload transported in one IP packet has a size of

$$P_v = \left\lceil \frac{N_F B_F}{8} \right\rceil \tag{5}$$

bytes, where $\lceil x \rceil$ is the smallest integer larger than or equal to $x$. Remark that not all payload sizes are possible. In RTP it is not allowed to split code words over packets. Hence, a voice payload always consists of an integer number of code words. Notice also that the voice payload consists of an integer number of bytes. Code words of the G.723.1 codec do not contain an integer number of bytes (see Table 1). For this codec, the code word corresponding to the bit rate of 5.3 kb/s (6.3 kb/s) is sometimes stuffed with 2 (3) bits to round them up to a code word of an integer number of bytes.

The service delay is the sum of the delays accumulated by putting the voice packet on the transmission link in every node. As mentioned before, we neglect the service delay in the backbone. The only remaining contributions to the service delay are the contributions of both access networks. To transport the payload $P_v$ over the IP network several headers are attached. The resulting IP packet size of a voice packet is given by

$$S_v = P_v + O_{IP/UDP/RTP} \tag{6}$$

bytes. Without header compression an RTP, UDP and IP header of respectively 12, 8 and 20 bytes are added, i.e. $O_{IP/UDP/RTP}$=40 bytes. In the case header compression is used this reduces to an overhead of just 2 bytes [3], i.e. $O_{IP/UDP/RTP}$=2 bytes. To transport the IP voice packets over the link in the access network further overhead is needed. In the case IP-over-PPP is used as transport protocol 7 more bytes are needed [8]. Hence, for each voice payload

$$F_v = S_v + 7 \tag{7}$$

bytes have to be transmitted. In the case IP-over-ATM [5] is used as transport protocol, first an AAL5 trailer of 8 bytes is added to the IP packet. As mentioned before no LLC header is necessary. Then the AAL5 frame is padded with the required number of bytes to make it a multiple of 48 before it is segmented in segments of 48 bytes. Finally, an ATM header of 5 bytes is attached to each segment. Hence, for each voice payload

$$F_v = \left\lceil \frac{S_v + 8}{48} \right\rceil 53 \tag{8}$$

bytes need to be transmitted. For transmitting $F_v$ bytes (over both access networks) a total service delay of

$$T_{serv} = 8\,F_v\left(\frac{1}{R_{up}} + \frac{1}{R_{down}}\right) \tag{9}$$

is required.

Before a voice payload can be put on the transmission link, it may need to queue. Again the backbone is assumed to be so fast that its contribution to the queueing delay is negligible. In both access networks there is only one voice application, which has to compete for the link capacity with all data applications (of the same user). Since voice packets get HOL priority over data packets the longest total queueing delay for a voice payload amounts to

$$T_{que} = 8\left(\frac{E_{up}}{R_{up}} + \frac{E_{down}}{R_{down}}\right) \quad , \tag{10}$$

in which $E_{up}$ and $E_{down}$ are the size of the interfering data chunks seen by an arriving voice payload in up- and downstream direction respectively. Table 2 gives the size of interfering data chunks for the considered access transport protocols.

| | File transfer | | Web browsing | |
|---|---|---|---|---|
| | $E_{up}$ (bytes) | $E_{down}$ (bytes) | $E_{up}$ (bytes) | $E_{down}$ (bytes) |
| ATM | 53 | 53 | 53 | 53 |
| PPP | 583* | 583* | 47** | 583* |
| F-PPP | 200 | 200 | 47** | 200 |
| S/R-PPP | 3 | 3 | 3 | 3 |

* = default IP packet size + PPP overhead = 576+7
** = acknowledgement packet size + PPP overhead = 40+7

**Table 2: Size of the interfering data chunks seen by an arriving voice payload in up- and downstream direction for different transport protocols and different data applications simultaneously active with the VoIP call.**

Some voice payloads of a voice flow may need to queue for the maximum queueing time given by eq. (10). Others need not to queue at all. Hence, the voice flow, which entered the IP network as a CBR* stream, is jittered when it arrives at the destination PC. Since the decoder also needs a CBR stream, a dejittering mechanism is necessary in the destination PC. The simplest dejittering mechanism consists of delaying the first payload of a voice flow over a delay $T_{jit}$ and then reading the following voice payloads out of the dejittering buffer at the constant rate of one voice payload every $N_F T_F$ ms. If the dejittering delay is chosen just equal to the difference between the maximal and minimal queueing delay, i.e. if

$$T_{jit} = T_{que} \quad , \tag{11}$$

every voice payload arrives in time to be read out. If the dejittering delay would be chosen smaller, a fraction of the voice payloads would arrive too late and would effectively be lost. If the dejittering delay would be chosen larger the timely delivery of the voice signal could be jeopardised. Remark that the voice signal undergoes the worst case mouth-to-ear delay if the first payload of the voice flow experienced the maximal queueing delay. The voice signal undergoes the best case mouth-to-ear delay if the first payload of the voice flow experiences no queueing delay. A slightly more complex, so-called adaptive dejittering mechanism operates on a talk spurt by talk spurt basis. This mechanism delays the first payload of every talk spurt over a dejittering delay that is adjusted from talk spurt to talk spurt based on the measured jitter characteristics of the past talk spurts. During each talk spurt the dejittering buffer is again read at a constant rate. We do not look at adaptive dejittering mechanisms in this paper, because a typical VoIP call will probably not last long enough to measure the characteristics of the jitter accurately enough.

Finally, the voice payloads experience several other delays. The most important contribution is the propagation delay $T_{prop}$ of 5 μs/km. Other contributions are the delay introduced by the echo controller, the delay caused by digitisation, route look-up delay in all IP nodes, switching delays, … Furthermore, some delays were neglected in the description of the five delay components above. For example the encoding $T_{enc}$ and decoding $T_{dec}$ that were previously neglected, in practice do take finite values, while the service delay will not be exactly proportional to the number of bits to be put on the transmission link (as expressed by eq. (9)), but there will be some offset to be added too. Crucial is that all these delays do not depend on the payload size and do not contribute to the jitter. Under these conditions the (small) impact of these delay components can be taken into account by artificially including them in the component $T_{prop}$.

## 4. Effective bit rate

The effective bit rate required for the transport of the voice signal is given by

$$R_{eff} = R_{cod} \frac{F_v}{P_v} \quad . \tag{12}$$

In this paper we only consider symmetric scenarios, i.e. we only study cases in which the access networks, access transport protocols and data applications simultaneously active with the VoIP call are the same for both calling parties. If the size of the interfering data packets and the bit rates of the access network are known, the required dejittering delay $T_{jit}$ can be calculated (see eqs. (10) and (11)). Furthermore, we suppose that both calling parties know what the propagation delay $T_{prop}$ is (or at least that they can estimate this propagation delay). We also assume that during call set-up the type of codec and the packet size are negotiated between the two calling parties.

There is some freedom in the choice of the number of code words $N_F$ packed in one voice payload $P_v$. There are two possibilities. First, the number of code words $N_F$ per payload can be chosen such that a mouth-to-ear delay of 150 ms is consumed as completely as possible. In that way the maximum quality corresponding with the considered codec is reached. Aiming for a lower mouth-to-ear delay, by choosing $N_F$ smaller does not make sense since it does not increase the quality, but increases the effective bit rate needed for the transport of the voice signal. Second, the number of code words $N_F$ per payload can be chosen such that the tolerable mouth-to-ear delay (as stated in Table 1 for each considered codec) is consumed as completely as possible. In that way the worst case VoIP call just attains traditional quality and the effective bit rate needed for the voice transport is as low as possible. In this paper we follow the second approach: we aim for the lowest possible effective bit rate.

There are two reasons why a VoIP call of traditional quality is not possible. The first reason is that the effective bit rate $R_{eff}$ exceeds the (up- or downstream) bit rate of the access network. The second reason is that even if payloads consisting of just 1 code word ($N_F$=1) are used for the transport of the voice signal, the tolerable mouth-to-ear delay bound is violated.

## 5. Results

All figures in this section express the effective bit rate as a function of the propagation delay. The effective bit rate is calculated under the assumption that the dejittering delay $T_{jit}$ is optimally set (see eqs. (10) and (11)) and that the number $N_F$ of code words per payload $P_v$ is chosen such that the tolerable mouth-to-ear delay (of Table 1) is consumed as much as possible. Hence, the effective bit rate is as small as possible. Normally the effective bit rate cannot decrease as the propagation delay increases. The reason is that the payload size cannot increase if the propagation delay increases, because if propagation consumes more of the tolerable mouth-to-ear delay, less remains for the packetisation and service delay. Due to the granularity of the codecs ($N_F$ is an integer) the payload $P_v$ is a step-wise constant function of the propagation delay $T_{prop}$. It is possible that at a certain propagation delay the payload size decreases to 0 (i.e. $N_F$=0). At that propagation delay the VoIP call can no longer attain traditional quality. We indicate this in the figures by putting $R_{eff}$=0 (although according to eq. (12) the effective bit rate is infinite).

Figure 2 illustrates the case of a high bit rate access network. Under these circumstances the service, queueing and dejittering delays are practically negligible compared to the tolerable mouth-to-ear delay. The method to determine the packet size described in Section 4 boils down to choosing the packet size such that the packetisation delay consumes as much as possible of what is left of the tolerable mouth-to-ear delay after subtraction of the codec delay and the propagation delay. Hence, the payload size is independent of the choice of the access transport protocol. Also in this case there is no danger that the effective bit rate of the VoIP call becomes larger than the (up- or downstream) access bit rate. Hence, a VoIP call of traditional quality can be supported even if a file transferring application is simultaneously active. IP-over-PPP as access transport protocol results in a smaller effective bit rate than IP-over-ATM, because for the same packet size IP-over-PPP introduces less overhead than IP-over-ATM (see eqs. (7) and (8)).

Figure 3 illustrates the case of low-speed access networks using traditional IP-over-PPP as transport protocol and with web browsing as the data application simultaneously active with the VoIP call. This figure shows that a VoIP call with the G.711 codec cannot attain traditional quality, because the effective bit rate it produces is larger than the available (upstream) bit rate in the access network.

Figure 4 isolates one low bit rate codec, i.e. the G.723.1 codec at 6.3 kb/s, and shows the influence of the data application simultaneously active with the VoIP call. In the case of a low-speed ADSL access network a VoIP call of traditional quality is always possible if the data application simultaneously active with the VoIP call is web browsing. If the data application is a file transferring application a long distance VoIP call ($T_{prop}$>25 ms) with traditional quality is not possible.  In the case of a N-ISDN access network a file transferring application makes that the VoIP call cannot attain traditional quality. If web browsing is the data application simultaneously active with the VoIP call, short distance calls ($T_{prop}$<25 ms) are still possible. Similar conclusions can be drawn for other low bit rate codecs.

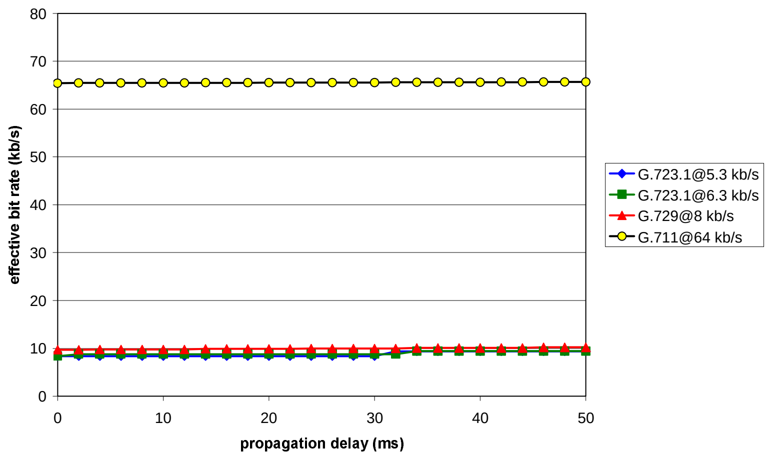



**Figure 2: The effective bit rate of a VoIP call as a function of the propagation delay for a high-speed ADSL access network using IP-over-PPP as transport protocol with a file transferring application simultaneously active with the VoIP call. Several codecs are considered.**

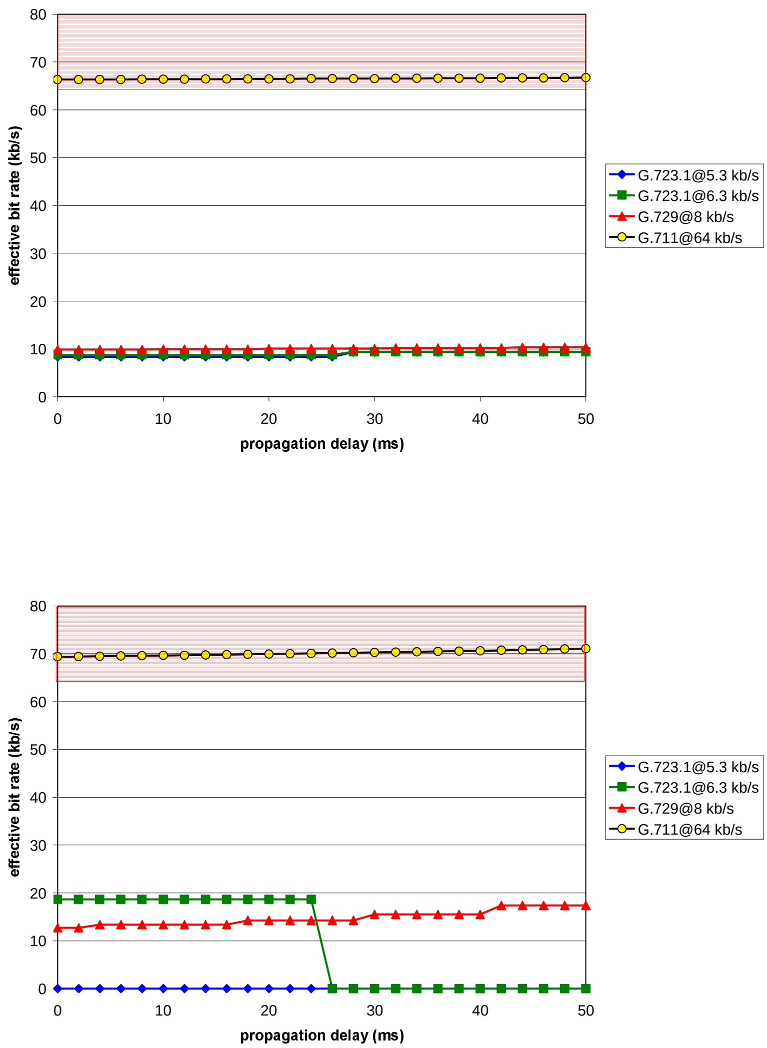
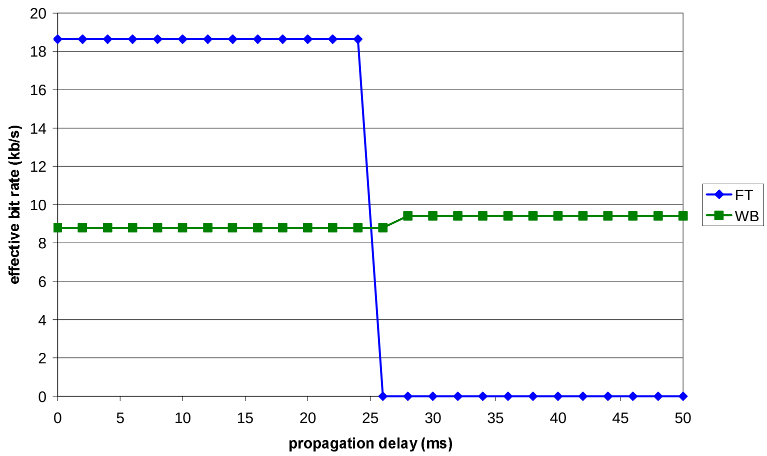



**Figure 3: The effective bit rate of a VoIP call as a function of the propagation delay for a low-speed ADSL (top) and N-ISDN (bottom) access network using traditional IP-over-PPP as transport protocol with a web browsing application simultaneously active with the VoIP call. Several codecs are considered.**



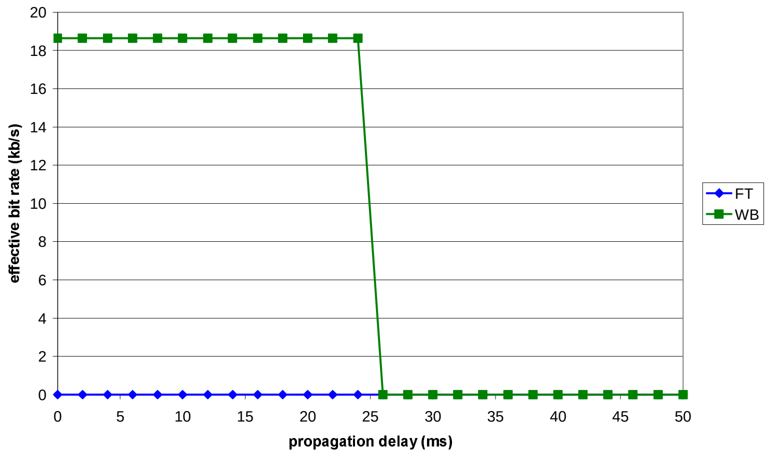
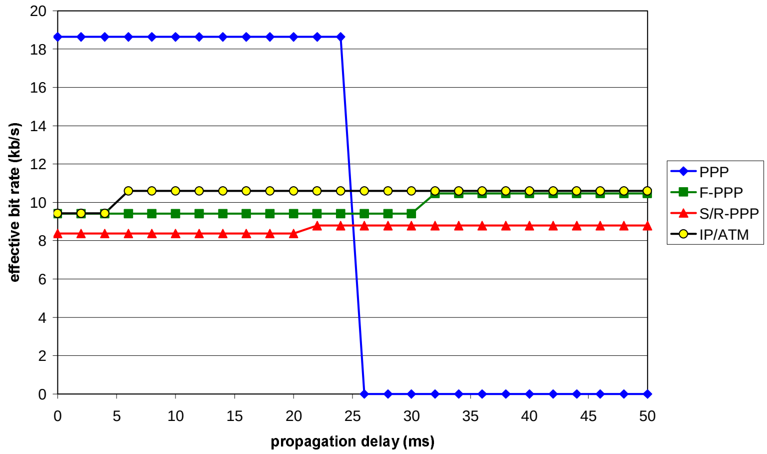




**Figure 4: The influence of the data application simultaneously active with the VoIP call on the effective bit rate of the VoIP call. The VoIP call using the G.723.1 codec at 6.3 kb/s is routed over a low-speed ADSL (top) or a N-ISDN (bottom) access network using traditional IP-over-PPP as transport protocol. WB indicates web browsing; FT indicates a file transferring application.**

Figure 5 illustrates the influence of the access transport protocol for a low bit rate access network. In the case the data application simultaneously active with the VoIP call impairs the quality of the VoIP call when using traditional IP-over-PPP, the use of an access transport protocol that segments the data packets into small pieces makes the VoIP call of traditional quality possible. Figure 5 shows that IP-over-F-PPP, IP-over-S/R-PPP and IP-over-ATM make a VoIP call of traditional quality simultaneous with a file transferring application possible, while traditional IP-over-PPP cannot guarantee this for all calls. S/R-PPP leads to the lowest bit rate. Which of the two other techniques, i.e. IP-over-ATM and IP-over-F-PPP, performs the best depends on the circumstances.



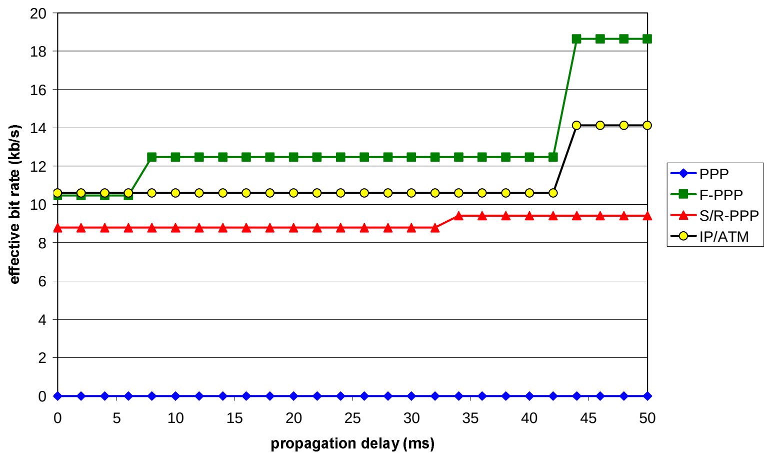
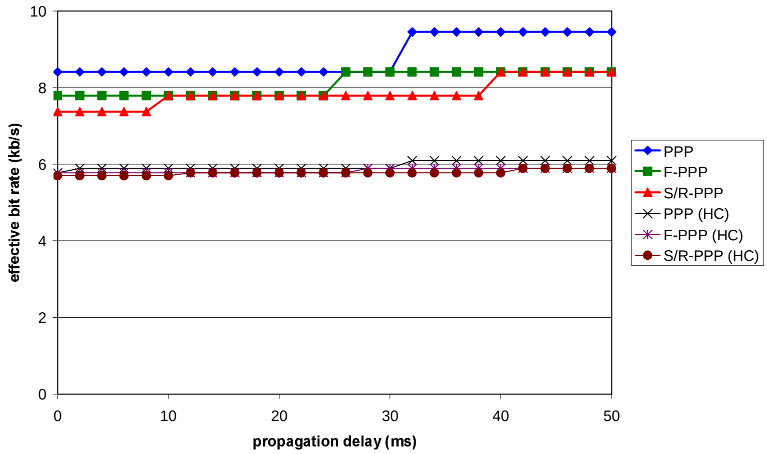




**Figure 5: The influence of the transport protocol on the effective bit rate of a VoIP call. The VoIP call using the G.723.1 codec at 6.3 kb/s is routed over a low-speed ADSL (top) or a N-ISDN (bottom) access network. The data application simultaneously active with the VoIP call is a file transferring application.**

Figure 6 illustrates the influence of header compression. The use of header compression has two possible influences. First, since less overhead bytes need to be sent over the access network, the service delay of an IP packet transporting a voice payload decreases. This means that for the same service delay the voice payloads can be chosen larger when header compression is used. This is only a minor effect. Second, as indicated by eq. (12) the effective bit rate in the access network decreases when header compression is used.

Header compression does not alleviate the problem when a VoIP call cannot attain traditional quality. The reason that a VoIP call cannot reach traditional quality is (almost always) the data application that occupies the server in the access networks for too long a time. Header compression on the voice packets does not alter anything to the data applications: the same interfering data packets still make a VoIP call of traditional quality impossible. Header compression can merely reduce the effective bit rate in the access network, if the VoIP call can already (i.e. without header compression) be conducted at traditional quality. It just reduces the amount of overhead bytes to be sent over the access network. Remark that in the backbone, where no header compression is possible, the effective bit rate is not altered.



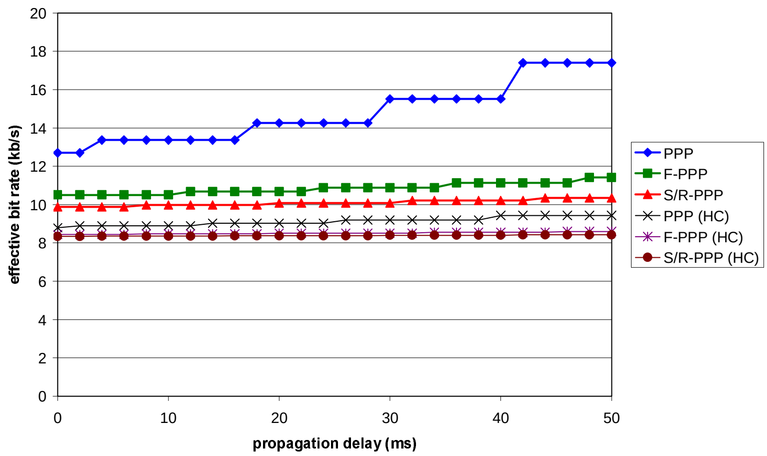




**Figure 6: The influence of header compression on the effective bit rate of a VoIP call. The VoIP call is routed over a high-speed ADSL access network with a file transferring application simultaneously active with the VoIP call using the G.723.1 codec at 5.3 kb/s (top). The VoIP call is routed over a N-ISDN access network with web browsing simultaneously active with the VoIP call using the G.729 codec (bottom).**

## 6. Conclusions

In this paper we studied VoIP calls routed over IP-based access networks to an IP backbone. We determined the impact of the choice of codec, the bit rates of the access network, the access transport protocol and the data application simultaneously active with the VoIP call on the effective bit rate required to conduct a VoIP call of traditional quality. We came to the following conclusions.

- For high access bit rates, we observe no problems, provided voice packets are given non-pre-emptive HOL priority over data packets in the access (and backbone) networks.
- If the minimum access bit rate is lower than or equal to 64 kb/s, the effective bit rate of the G.711 codec is too large. Hence, a low bit rate codec is required in this case.
- If the use of a low bit rate codec still does not resolve the problem, there are several possibilities. When using IP-over-PPP, the first solution is to avoid a file transferring application simultaneously active with the VoIP call, but allow only web browsing during the VoIP call. This is most suitable on asymmetric access networks. The second solution is to use a fragmentation technique: F-PPP or S/R-PPP. S/R-PPP always outperforms F-PPP. In that case it does no longer matter if the data application is a file transferring application or web browsing. The third solution is to use IP-over-ATM, although it is always outperformed by IP-over-S/R-PPP.
- For all access transport protocols that are able to meet the tolerable mouth-to-ear delay bound, header compression can be used to decrease the effective bit rate of the VoIP call in the access network.

## Abbreviations*

| | | | |
|---|---|---|---|
| AAL5 | ATM Adaptation Layer | N-ISDN | Narrowband Integrated Services Digital Network |
| ADSL | Asymmetric Digital Subscriber Line | PPP | Point-to-Point Protocol |
| ATM | Asynchronous Transfer Mode | PSTN | Public Switched Telephone Network |
| CBR | Constant Bit Rate | QoS | Quality of Service |
| F-PPP | Fragmentation-oriented PPP | RTP | Real-time Transport Protocol |
| GSM | Global System for Mobile communications | S/R-PPP | Suspend/Resume-oriented PPP |
| HOL | Head-Of-Line | UDP | User Datagram Protocol |
| IP | Internet Protocol | VC | Virtual Channel |
| LLC | Logical Link Control | VoIP | Voice over IP |

## Acknowledgments

This work was carried out within the framework of the project LIMSON sponsored by the Flemish Institute for Science and Technology (IWT).

## References

[1] C. Bormann, “The multi-Class Extension to Multi-Link PPP”, IEEE draft draft-ietf-issll-isslow-mcml-04.txt

[2] C. Bormann, “PPP in real-time oriented HDLC-like framing”, IETF draft draft-ietf-issll-isslow-rtf-03.txt

[3] S. Casner, V. Jacobson, “Compressing IP/UDP/RTP Headers for Low-Speed Serial Links”, IETF RFC 2508, Feb. 1999.

[4] D. De Vleeschauwer, J. Janssen, G.H. Petit, “Delay Bounds for Low Bit Rate Voice Transport over IP Networks”, SPIE International Symposium on Voice, Video and Data Communications, Boston, 19-22 Sept. 1999. (to appear)

[5] J. Hienanen, “Multiprotocol Encapsulation over ATM Adaptation Layer5”, IETF RFC 1483, July 1993.

[6] T.J. Kostas, M.S. Borella, I. Sidhu, G.M. Schuster, J. Grabiec, J. Mahler, “Real-Time Voice over Packet-Switched Networks”, IEEE Network, pp. 18-27, Jan./Feb. 1998.

[7] E.E. Mier, “VoIP Gateways—Tradeoffs Affecting Voice Quality”, BCR Voice 2000, Jan. 1999.

[8] W. Simpson, “PPP in HDLC-like framing”, IETF RFC 1662, July 1994.

[9] K. Van Der Wal, M. Mandjes, H. Bastiaansen, “Delay Performance Analysis of the New Internet Services with Guaranteed QoS”, Proceedings of the IEEE, Vol. 85, No. 12, pp. 1947-1957, Dec. 1997.