CURRENT TRENDS IN NETWORK MANAGEMENT K.G. Saur Verlag, München, 2007

Chr. Hoene, K. Clüver, and J. Weil

An Architecture for a Next Generation VoIP Transmission System

Dr. *Christian Hoene* is a postdoc at the "Computer Networks and Internet" group at the University of Tübingen, Germany. He is a researcher covering areas about Internet based voice communication, wireless transmissions, location tracking, security, accounting, and charging. Christian has studied at Technical University of Berlin in the TKN Group. He is the author of about 20 publications, 1 ITU standard contribution, and received one best paper award. Herr Hoene was also awarded in 2000 with Erwin Stephan Prize for passing his studies of computer engineering at the TU-Berlin with distinction. In 2001, he was a visiting scholar at the COMET Group of Prof. Campbell, Columbia University, New York. In 2005 he finished his Ph.D. studies on voice over WLAN also with distinction. Since 2007 he is a supported within the elite programme for postdocs, which is funded by the Landesstiftung Baden-Württemberg.

Kai Clüver; Studium der Elektrotechnik an der Technischen Universität Berlin, anschließend wissenschaftlicher Mitarbeiter am damaligen Institut für Fernmeldetechnik; 1998 Promotion; Industrietätigkeiten 1998 bis 2000 bei der PSI AG, Velbert, und 2000 bis 2003 bei der Cortologic AG, Berlin; zur Zeit wissenschaftlicher Mitarbeiter am Fachgebiet Nachrichtenübertragung der TU Berlin sowie Lehrbeauftragter an der TU Berlin und an der Technischen Fachhochschule Berlin; Lehr- und Forschungsschwerpunkte: Sprach- und Audiosignalverarbeitung und -übertrgaung, Sprachcodierung, Spracherkennung.

Jan Weil studierte bis 2004 Elektrotechnik an der Technischen Universität Berlin. Während seines Studiums arbeitete er als studentischer Mitarbeiter am Heinrich-Hertz-Institut an der Optimierung von Sprachcodecs für eingebettete Systeme. Seit 2005 ist er wissenschaftlicher Mitarbeiter am Fachgebiet Nachrichtenübertragung der Technischen Universität Berlin. Seine Forschungsinteressen liegen in den Bereichen Sprach- und Audiosignalverarbeitung und -codierung sowie Music Information Retrieval.

1 ABSTRACT

Packetized speech transmission systems implemented with Voice over IP are gaining momentum against the traditional circuit switched systems despite the fact that packet switched VoIP is two to three times less efficient than its circuit switched counterpart. At the same time, it only supports a rather bad "toll" quality. We believe that it is time for a new architecture developed from scratch – an architecture that includes an Internet enabled speech codec and its transport system. This architecture manages the perceptual service quality while using the available transmission resources to its best. The transmission of speech is managed and controlled with respect to its speech quality, mouth-to-ear delay, bit-rate, frame-rate, and loss robustness. Beside the architecture, we describe the requirements for the Internet speech codec and its transport protocol and present an interface between the speech codec and the transport protocol.

2 INTRODUCTION

Internet Telephony is a mature technology that has gained increasing popularity against the traditional Public Switched Telephone Network (PSTN) systems. Voice over IP (VoIP) is replacing the PSTN service on broadband access networks such as cable modems and DSL, as it is more cost efficient to also use IP broadband access for Internet telephony. In addition, future wireless broadband access networks such as the 3GPP's Long Term Evaluation (LTE) radio technology will support telephone services only via VoIP [1].

Despite the success, Internet Telephony has a fundamental drawback. It is much less bandwidth efficient than its classic circuit switched counterpart. VoIP requires two to three times more physical gross bandwidth than a modern circuit switched speech transmission in DECT, GSM, or UMTS networks. If more bandwidth is required, other performance parameters are to be sacrificed, too: the typical talk time of a mobile, portable VoIP telephone is shorter in comparison to cellular phones because more energy is required to support the transmission of packetized voice. This usually also applies to its transmission range.

If we compare commercial, modern mobile and cordless phones, one can see that a DECT telephone using circuit switched technologies has a talk time at least three times longer than a WLAN cordless phone – assuming similar battery capacities. Furthermore, using the circuit switched GSM technology, the transmission range is 10 to 100 times larger than using VoIP-WLAN technology, again assuming the telephones have the same battery capacities and talk times[1].

Taking these facts in consideration, one can say that a circuit switched based telephone call is far more efficient than its VoIP-WLAN counterpart. Because other portable VoIP based phones have similar operational specifications as well, we believe that the lack of efficient transmissions of the current VoIP architecture is fundamental and valid, regardless of any implementation details and product models.

Traditionally, VoIP uses speech compression schemes, which have been designed with circuit switched telephone systems in mind, such as ISDN or GSM, and have static frame rate and packet loss robustness. In the Internet, many more transmission parameters need to and can be controlled and managed. These include – beside the bit rate of the speech coder – the frame and packet rate, the loss robustness, and the algorithm delays. We believe that it is necessary to develop both a speech codec and a transport protocol which are optimized for the path characteristics of the Internet. They should be aware of the current transmission resources and the perceptual quality of the ongoing telephone call in order to adapt their transmission parameters autonomously.

Recent research results, which we will refer to in the following sections, have shown that current VoIP systems can indeed be significantly enhanced, both in terms of efficiency and quality. To gain efficiency we cannot be backward compatible nor support the classic speech coders or transport protocols such as ITU G.729 or IETF RTP. Instead, we need to break with the past and make a new start. If one took the freedom to design a new VoIP system from scratch, what would it look like?

In the following section we will propose a new architecture and describe how to develop an efficient speech transmission system including a speech coding framework and a transport protocol. We will also describe the motivation behind our design decisions referring to previous research results. In section 4 we will go into details and describe an interface between the speech codec and the transport protocol, explaining which parameters are exchanged. Finally, we will give an outlook on the upcoming design and implementation of the new speech codec, which will be optimized for the Internet, and its corresponding transport protocol.

3 ARCHITECTURE

The next generation VoIP architecture should consist of a speech codec, optimized for the Internet, and a corresponding transport protocol. The transmission should be bidirectional as telephone calls are bidirectional as well. Fig. 1 depicts the architecture's components. For a better overview, only one side of the communication path is shown. In the following, we describe the components individually.

3.1 Quality of the Telephone Call

In order to optimize the transmission of the telephone call perceptual quality models, which simulate the human rating of the quality of telephone calls, should be applied. The foremost quality model to mention is the ITU E-model which is intended as a planning instrument for telephone systems [5]. It considers most of the parameters that have an effect on the transmission quality, such as loudness of speech signal, noise levels, loudness of echoes, speech quality, and acoustic mouth-to-ear delay. It calculates an overall quality rating called the R factor that ranges from 0 (worse) to 100 (very good). Beside its primary purpose to plan transmission systems, it can also be applied in real time to control a transmission and set the various transmission parameters [6].

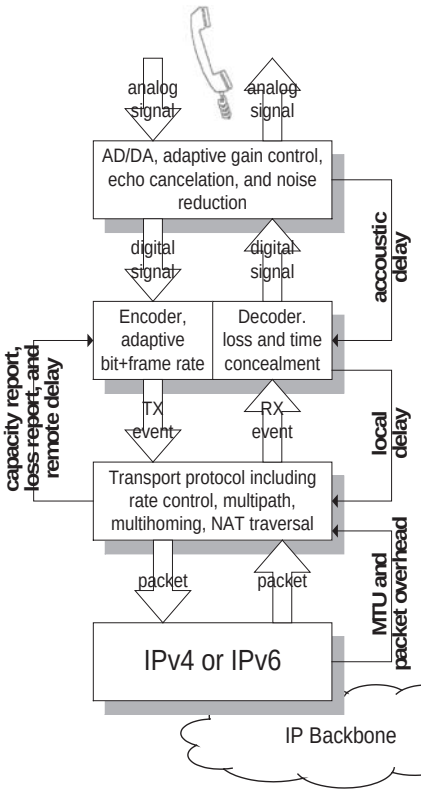

Fig. 1 Architecture for a Next Generation VoIP transmission system.

In the novel VoIP architecture, a quality model similar to the E-Model is of the utmost importance as it gives an overview on which parameters need to be optimized to achieve a high transmission quality. Additionally, a trade-off between speech quality and delay will be possible.

We can also derivate the first building blocks of the architecture, namely adaptive gain control (AGC), acoustic echo cancellation (AEC), and the determination of the intrinsic delay of a telephone, which are the sum of all delays that the telephone adds to the overall mouth-to-ear delay. In order to properly approximate the mouth-to-ear delay, the telephone should determine the intrinsic latency of the speech signal. For example, the AEC can be used to determine this delay.

3.2 Speech Codec and Concealment

In the last years many speech codecs, comprising of speech encoder, speech decoder, and loss concealment algorithms, have been developed and applied in PSTN, cellular networks, and VoIP networks. The speech codecs include ITU G.711, ITU G.729, Speex, ETSI GSM-EFR, 3GPP AMR, 3GPP AMR-WB,

1 These statements are based on a comparison of the specifications of commercial phones. As an exemplary DECT based cordless phone, we have chosen the Siemens Gigaset S44, which comes with a battery of 750 mAh, has a talk time of 10 h, and has a transmission range up to 300 m. As an example for both GSM and VoIP-WLAN we take the Nokia E70 model, which has a battery capacity of 970 mAh. In the GSM mode, it has a talk time between 3.3 and 6.4 hours and a transmission range up to 35 km. In the VoIP/WLAN mode using IEEE 802.11g it has a talk time between 3 and 3.2 hours and a transmission range similar to the DECT phone.

3GPP2 VMR-WB, and IETF iLBC. They have been optimized to provide superior speech quality, low algorithmic delay, low computational complexity, and high packet loss robustness. At the same time, they require a low transmission bit rate. Thus, why should we consider the development of new speech coders if the existing ones are perfect?

Three arguments, based on recent research results, have given us the insight that the current, standardized speech codecs might not be perfectly matched for the requirements of the Internet. The first one is based on the observation that the loss of speech frames can have quite a different impact on the speech quality and that many low rate speech codecs still allow a high loss rate without a perceptible degradation of the speech quality. The second one is based on the observation that low bit rate is not the only transmission parameter that is of importance in a packetized network. The third argument simply accounts for the observation that telephones are not only used for human to human conversation but also increasingly frequently to listen to and to exchange music.

3.2.1 The Unequal Impact of Losing Speech Frames

For a long time it has been known that the impact of speech frame losses can differ widely. Some losses, even during voice activity, are hardly perceived. Others have a notable negative impact on the speech quality. Just recently, one of the authors has investigated this effect systematically [7]. A measurement procedure has been developed to quantify the impact of single packet or speech frame losses. This measurement procedure has been verified by formal listening-only tests to ensure its precision. Moreover, a metric has been developed that quantitatively describes the impact of losses on speech quality.

Using the importance of speech frames, simulation and listening tests show that many speech frames can be dropped during voice activity because the loss concealment on the receiver side works so well that the losses are hardly notable [7]. These studies were conducted for G.711, G.729, and AMR encoded voice data and loss rates up to one third (during voice activity) still allow understandable speech transmissions. Thus, knowing the importance of speech frames, significant performance gains can be achieved if only important packets are transmitted.

As a result of these research studies, one can say that the speech coders under study still contain a high level of redundancy because many speech frames need not to be transmitted (or can be dropped intentionally). Furthermore, the information about the speech is unequally distributed among the speech frames as some frames are important and others are not. Would it not be better if all speech frames had the same importance and all speech frames contained the same amount of information? Then, each packet loss would have a similar impact on the degradation of speech quality.

This can only be achieved if the size of the speech frames is variable (such as in the 3GPP2 VMR-WB speech codec [3]) or if the rate of frames varies over time. Then, if the current speech signal contained a lot of new information, the encoder would produce larger or more speech frames, otherwise the encoder would produce smaller or less speech frames[2].

2 Indeed, the AMR's discontinuous transmission (DTX) algorithm produces smaller and less frequent speech frames during silence. However, during voice activity the frames have all a constant size and are produced every 20 ms.

We assume that future speech codecs, optimized for the Internet, will generate speech frames of similar importance. The speech codecs will have variable frame size and/or variable frame rates.

3.2.2 Bit Rate and Frame Rate

Many speech codecs of today support multiple bit rates. For example, the AMR codec supports eight compression rates ranging from 4.75 to 12.2 kbps. Others, like the Speex codec, support a bit rate range from 2.15 to 44.2 kbps. If a highly efficient transmission is to be achieved in a VoIP system because, for example, bandwidth or energy is scarce, then often the lowest bit rate is chosen. A low bit rate has low bandwidth requirements and fewer bits per second need less transmission energy.

Again, recent research results have shown that the bit rate is not the only factor that influences the transmission efficiency: The packetisation can be of equal importance. Packetisation determines how many speech frames, produced by the speech encoder, are put into a VoIP packet before the packet is transmitted. Many speech coders produce one frame per 10, 20, or 30 ms. Many VoIP telephones transmit those frames in VoIP packets every 20, 40, or 60 ms. Thus, one VoIP packet contains one or multiple speech frames.

If more speech frames are put into one VoIP packet, a longer time has to be waited before the VoIP packet can be transmitted. Thus, the algorithmic delay of the packetisation increases. On the other hand, if less VoIP packets are transmitted per second, the gross bandwidth is reduced because less protocol headers such as IP, UDP, and RTP need to be transmitted.

If now the bandwidth is limited, should the coding rate be reduced or the packetisation be increased in order to save bandwidth? Simulations have been conducted in [7] to answer this question. The results show that the answer to this depends on the underlying technology. On a traditional circuit switched connection, which does not transmit packet headers, the reduction of the bit rate achieves the best quality. On switched Ethernet links using an AMR codec, both bit and packet rate should be adapted. And, finally, on an IEEE 802.11b wireless LAN using an AMR codec, it is sufficient to decrease only the packet rate to save a significant part of the bandwidth.

The results also show that Internet optimized speech should not only support a low and variable bit rate. The frame rate is of similar importance. This means, a speech coder should not produce frames at a constant rate but should reduce the packet rate, whenever this is possible without sacrificing the perceptual service quality.

If then the speech coding implemented variable frame rate, it would not be necessary to include multiple speech frames in one VoIP packet. Instead, the encoding would generate one speech frame of the appropriate length. Thus, only one speech frame will be transmitted in one VoIP packet.

We believe that the Internet optimized speech coders should be able to produce speech frames at any point of time. For example, speech frames can be generated if the current change of speech characteristics requires to do so. An Internet speech codec must not follow the strict rule of a constant time interval. Recent research results have shown that a speech codec with optimal time-segmentation and thus packet size can be indeed developed [11][12]. However, these coding technologies are still in their infancy.

3.2.3 Limitations of the Frequency Band

Quite frequently it can be seen that mobile phones are not only used for human to human communications but for many other purposes like listening to music, exchanging ring tones, listening to the radio, and many more. We assume that, in the future, telephones will also be required to transmit musical content.

Current speech codecs are intended for the transmission of human speech (and background noise). Recently, enhancements, such as 3GPP's AMR-WB+, the AAC-Low delay, and Fraunhofer's Ultra Low Delay (ULD) codec, support the transmission of music in real time. However, current VoIP telephones use codecs that support either a "narrow" frequency bandwidth up to 3700 Hz or a "wideband" frequency bandwidth up to 7000 Hz. But, in contrast to the traditional PSTN or cellular systems, VoIP has no technical constrains that limit the frequency spectrum. Instead of this, an Internet speech codec should encode speech and music at the highest quality that the current transmission path can support to transmit.

3.2.4 Loss and Time Concealment

Packet loss concealment algorithms are placed at the receiving end of a transmission of speech and limit the effect of packet losses [9]. They extrapolate the last part of the speech signal if the current speech frame has not been received. In this manner they limit the negative effect of packet losses on the speech quality. Nowadays, they are often part of a speech codec's standardization document and part of the decoder.

Time concealment tries to cope with the effect of transmission jitter by slowing down or increasing the speed of the current speech [10]. Time concealment algorithms have a positive effect on the service quality, but they come at the cost of additional algorithmic delay. Additionally, if a speech frame has not been received on time, the decoder cannot decide whether to slow down the speech output or whether to conduct loss concealment. At this moment of time, the decoder cannot know whether the packet will still arrive or whether is has been lost.

On the other hand, if the decoder closely followed the delay process of the transmission path, the overall mouth-to-ear delay could be reduced significantly. The buffering of speech frames in a play out buffer, nowadays included in nearly all VoIP phones, could be omitted. Thus, we suggest to include the loss concealment, the time concealment, and the playout buffer in the decoder. The decoder should then decide to play back the speech frames as they arrive and conceal, slow down, or fasten the speech, if required.

3.3 Transport Protocol

The Internet optimized speech codec should not operate on the traditional RTP/UDP protocol. Instead, it requires a transport protocol which provides all information on the current state and quality of the transmission path. Only if the speech codec knows the current properties of the transmission path its coding bit rate and packet rate can be adapted to achieve a high perceptual transmission quality.

Forward Error Correction (FEC) should not be a functionality provided by the transport protocol. It can be implemented more easily at the encoder. But then the transport protocol should inform the encoder about the loss process in the network and the encoder should adapt its loss robustness.

The transport protocol should take advantage of the bidirectional nature of a telephone call and transmit speech frames bidirectionally. Thus, control information, nowadays transmitted in signaling packets like RTCP, can piggyback on the data stream. By this means, the packet rate can be further reduced. In addition, the transport protocol can implement feedback loops to control rate and congestion more easily. Optionally, the transport protocol can support other mechanisms such as multihoming, mobility, multipath, or NAT traversal in order to increase the reliability and quality of the transmission.

4 INTERFACE DESCRIPTION

After the description of the architecture, this chapter depicts a possible interface between the speech codec optimized for the Internet and its corresponding transport protocol. This interface description is required, if both speech codec and transport protocol are to be developed separately or if codecs or transport protocols should be exchangeable.

In this publication we are concentrating on continuous transmission of speech. State changes are notified by events. Events change parameters and data between the codec and the transport protocol. To describe the parameters that are exchanged between both entities, we use a Java-like pseudo code notation.

4.1 Coding to Transport: Transmit Event

The speech coder notifies the transport layer every time a new frame has been generated. Beside the frame data, its length, and time stamp is required. The length and time stamp can both be dynamic because the speech coder might have a variable speech and coding rate (such as the proprietary codec iSAC from Global IP Sound and 3GPP2's VMR-WB).

```
class TransmitEvent {
   byte data[]; // speech frame and its length
   int ts;      // time stamps defining when the speech signal
                   as been produced (local clock)
};
```

Time stamp is a novel feature but an important one because one cannot assume that speech frames are produced at regular intervals. Also, the time stamp should be taken at the point of time the speech signal has been spoken or produced.

Given this information, the transport layer can calculate the current bit and frame rates generated by the encoding. Given a set of transmit events called $te[1]$ to $te[n]$ all time stamps should be increasing. That means, for all $1 = i < n$, $te[i].ts < te[i+1].ts$. Then, bit rate and the packet rate are calculated as

$$bitrate = \frac{8 \cdot \sum_{i=1}^{n} te[i].data.length}{te[n].ts - te[1].ts} \quad (1)$$

$$packetrate = \frac{n}{te[n].ts - te[1].ts} \quad (2)$$

The main task of the transmission layer is to transmit the frame data, its length, the time stamp, and its increasing index. These parameters should be transmitted to one (or multiple) destinations. How the transport layer opens and tears down its connection and whether the transport layer uses multiple destinations to support multicast, multiple paths, or any kind of error correction is beyond the scope of this publication.

A second task is to estimate the variability of the flow of speech frames. Depending on the current situation of the conversation, the variability of speech on the one side and the interactivity on the other side can vary significantly. Thus, the rate and size of speech frames can differ substantially. The transport protocol requires an estimate of the variability of transmission rates in order to calculate a safety margin regarding the transmission capacity.

4.2 Transport to Decoding: Receive Event

The transport protocol again hands over speech frames to the decoder as soon as it receives them. It should not buffer the speech frames. The data parameter includes:

```
class ReceiveEvent {
   byte data[]; // speech frame and its length
   int ts;      // time stamps defining when the speech has been
                   spoken (remote clock)
   int jitter;  // time offset as compared to mean remote round
                   trip time describe in section 3.3.
   short index; // increasing index number of the speech frame
};
```

The receiver calculates the loss rates using a set of receive events called $re[1]$ to $re[n]$, where for all $1 = i < n$, $re[i].ts < re[i+1].ts$ and $1 \leq i < n$, $re[i].index < re[i+1].index$:

$$packetlossrate = \frac{n}{re[n].index - re[1].index} \tag{3}$$

Furthermore, using the time stamps, the decoder can calculate the transmission delay variations. This allows the decoder to get a statistics about the distribution of the transmission delays in order to adapt the play out of the speech frames accordingly.

4.3 Transport and Codec: Round Trip Times Delays

The classic RTP Control Protocol (RTCP) is a signaling protocol to provide feedback on the quality of the transport of multimedia data. The feedback is performed using the RTCP sender and receiver reports, which report information about time stamps in regular intervals, byte- und packet counts, loss rates, smooth mean deviation of inter-arrival times (jitter), and the round trip times [2].

Recently, Extended Reports (XR) have been added to RTCP to report more detailed statistics on the network characteristics or quality monitoring [8]. The data provided includes which packets have been lost and received, which packets have been received multiple times, and when the packets have been received. Additionally, it provides the means to gather the network round trip time and the end system delay in order to calculate the acoustic round trip time.

As mentioned above, the mouth-to-ear delay is an important quality metric that influences the service quality of a telephone call, and needs to be optimized. More precisely, the metric under optimization is the acoustic round trip time, which is the sum of the mouth-to-ear delays of both transmission directions. Humans cannot distinguish which direction of the transmission contributes to the delay, thus the one-way delay needs not to be known.

The round trip time can be used for both the codec and the concealment. For example, if the RTT is below 150 ms, the codec increases its algorithmic delay to better cope with packet loss or with delay variations.

Both the codec and the transport protocol inform each other, if the mean acoustic delay of each side has changed. The following event format is applied:

```
class RTTchange {
   int delay;  // acoustic round trip time on the local or
                  remote side
};
```

The sum of both values, from the codec and the transport protocol, is the overall acoustic round trip time and twice the mean mouth-to-ear delay. The events are triggered only if the delay has changed significantly, e.g. more than about 10 ms, to avoid an unnecessarily high number of updates.

4.4 Transmission Capacity

The transport protocol determines the rate at which the coder is allowed to produce data. It informs the codec about this rate. In compliance with TCP, the rate is given in bits per round trip time, which means that the coder is allowed to send up to the number of bits within the next round trip time. The coder is free to choose when it sends the data, either at the beginning, continuously during, or at the end of the RTT period. The capacity of the path can change highly dynamically. Thus, an update regarding the transmission rate can occur at any time.

Depending on the volatility of the coder's rate and the volatility of the network bandwidth, the transport protocol is free to reduce the transmission rate to add a safety margin or to increase the transmission rate in order to achieve a statistical multiplexing gain at the cost of a higher packet loss rate.

TCP sends packets at the maximal transfer unit (MTU) in order to achieve the highest throughput. If a service required a low transmission delay, it would not benefit from sending large packets containing a long speech segment but from short packets containing short speech segments.

Usually, the cost of sending many small packets is much higher than sending one larger packet, because each packet has additional packet headers on multiple layers. In addition, the medium access control requires additional resources to transmit a packet.

An example given in [7] studied the transmission over IEEE 802.11b at 11 Mbps in the DCF mode. The cost of the contention period, collisions, and the immediate acknowledgements contribute significantly to the bandwidth requirements of a packet, beside the headers of PLCP, MAC, Link-layer protocol (IEEE 802.3), IP, UDP, and RTP. In total, transmitting one packet, the physical medium of IEEE 802.11b is busy for about one microsecond in addition to the actual data transmission.

Thus, the costs of one packet – regardless of its size – correspond to about 1000µs/11MBps ≈ 1500bytes/s in the IEEE 802.11b mode. Packet headers can be easily compressed to a few bytes by using the IETF IP header compression algorithms. But header compression cannot reduce the overhead of the MAC and link layer protocol.

In [4], the notion of packet overhead is introduced to determine the amount of overhead required to transmit a packet. It is defined as the gross bandwidth that is required to transmit a packet

$$t_{overall} = t_{overhead} + \frac{ps_{pdu}}{rate} \quad \Leftrightarrow \quad t_{overall} \cdot rate = t_{overhead} \cdot rate + ps_{pdu} \quad (4)$$

with ps_{pdu} being the packet size of the PDU.

Defining $p_{overhead}=t_{overhead}$.rate, the packet overhead is the number of bytes that each packet costs. It measures the gross number of bits on the physical medium.

Of course, this value can change with the physical medium, the transmission rate, and many other parameters. If the packet overhead is not precisely known, the transport protocol can guess it by averaging the packet overhead of various, typical, and commonly used transmission technologies. For this interface description, we apply the notation of packet overhead: The transport protocol signals the coder the current transmission requirements as

```
class Capacity {
   int bps;       // mean bit per second the coder is allowed
                     to produce at maximal during the next
                     round triptime.
   int mtu;       // the maximal transfer unit, the largest
                     packet size a coder is allowed to produce
   int overhead;  // costs of a single packet in bits
};
```

Thus, for the transmit events $i = \in \{1;n\}$ within a period of t_{rtt}, the following conditions must be given:

$$te[i].data.length \le capacity.mtu \quad (5)$$

$$capacity.bps \cdot 8 \cdot t_{rtt} \ge \sum_{i=1}^{n} \left(te[i].data.length + capacity.overhead \right) \quad (6)$$

4.5 Transport to Coder: Packet Losses

In the Internet, packet losses occur during time of congestion. Furthermore, on wireless links transmission errors might cause packet losses. Following the solution given in the RTCP XR receiver reports [8] we report packet losses and packet receptions using a bit vector.

```
class PacketLossReport {
   short begin_index;  // the first index number that this event
                          reports on
   short end_index;    // the last sequence number that this
                          event report on plus one.
   int vector[];       // the array of integers is read from
                          left to right, in order of increasing
                          index number
                       // (with the appropriate allowance for
                          a wraparound)
};
```

The coder requires the report about packet losses to adapt its loss robustness and change the amount of redundancy. If many losses occur, the amount of redundancy should be increased to help the packet loss concealment algorithm. But if the losses held on for a long time and were bursty, redundancy could not help and losses would be inevitably audible.

5 SUMMARY AND OUTLOOK

We followed the following tenets in our architectural redesign of a VoIP transmission system:

1. Develop a speech codec that supports a variable bit as well as a variable frame rate.
2. Closely couple the speech codec and the transport to achieve the benefits of a cross layer optimization strategy. They should be aware of the current quality of the call in order to control their transmission parameters.
3. Include Forward Error Correction in the encoder.
4. Combine decoding, loss and time concealment, and the playout buffer into a single Internet enabled speech decoder.
5. Do not stick to a narrow or wide frequency band because, beside speech, also music transmission will be required.

This publication is meant to help researchers design and implement a new architecture for the next generation of VoIP transmission system. But not until this system has been designed, implemented, and tested, we can see to what extent the new architecture can enhance the transmission efficiency and perceptual quality as compared to the classic VoIP system.

6 REFERENCES

[1] 3GPP TR: “Feasibility study for evolved Universal Terrestrial Radio Access (UTRA) and Universal Terrestrial Radio Access Network (UTRAN),” version 7.1.0, Oct. 2006.

[2] H. Schulzrinne; S. Casner; R. Frederick; V. Jacobson: “RTP: A Transport Protocol for Real-Time Applications”, IETF RFC 3550, Jul. 2003.

[3] 3GPP2 C.S0052-A v1.0: “Source-Controlled Variable-Rate Multimode Wideband Speech Codec (VMR-WB) Service Options 62 and 63 for Spread Spectrum Systems”, 3GPP2 Technical Specification, Apr. 2005.

[4] C. Mahlo; C. Hoene; A. Rostami; A. Wolisz: Adaptive Coding and Packet Rates for TCP-Friendly VoIP Flows, Proc. 3rd Int. Symp. on Telecommunications (IST2005), Shiraz, Iran, Sep. 2005.

[5] ITU G.107: “The E-model, a computational model for use in transmission planning”, Mar. 2005.

[6] C. Hoene; H. Karl; and A. Wolisz: “A perceptual quality model intended adaptive VoIP applications”, International Journal of Communication Systems, Wiley, Aug. 2005.

[7] C. Hoene: “Internet Telephony over Wireless Links”, PhD thesis, Technical University of Berlin, TKN, Dec. 2005.

[8] T. Friedman; R. Caceres; A. Clark: “RTP Control Protocol Extended Reports (RTCP XR),” IETF RFC 3611, Nov. 2003.

[9] K. Clüver; and P. Noll: Reconstruction of missing speech frames using sub-band excitation. In: IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis, 1996.

[10] Y.J. Liang; N. Färber; and B. Girod: “Adaptive playout scheduling and loss concealment for voice communication over IP networks,” IEEE Transactions on Multimedia, vol. 5, no. 4, pp. 532-543, Dec. 2003.

[11] P. Prandoni; M. Vetterli: “R/D optimal linear prediction”, IEEE Transactions on Speech and Audio Processing, Vol. 8, Iss. 6, Nov. 2000.

[12] C.A. Rodbro; J. Jensen; R. Heusdens: “Rate-distortion optimal time-segmentation and redundancy selection for VoIP”, IEEE Transactions on Audio, Speech and Language Processing, Vol. 14, Iss. 3, May 2006.