Voice Over IP: Improving the Quality Over Wireless LAN by Adopting a Booster Mechanism - An Experimental Approach[1]

Christian Hoene, Iacopo Carreras, Adam Wolisz
Telecommunication Networks Group
Technische Universität Berlin
{hoene, carreras, wolisz}@ee.tu-berlin.de

ABSTRACT

The performance of unreliable voice transmission (Voice over IP) over wireless links is measured not by the throughput but by the perceptual speech quality. The speech quality is impaired by packet losses, which are common on wireless links, and by high transmission delays. In this paper, we describe the design and implementation of a novel *Speech Property Based Booster* that improves the quality of voice over wireless LANs. This booster is in compliance with existing standards and is transparent to other protocols. It uses characteristics of human speech production and features of modern audio codecs to differentiate packets regarding their importance for perceptual quality. Important packets are protected at the link layer by three mechanisms: *selective packet loss recovery, redundant transmission* and a *hybrid solution*. These mechanisms have been evaluated using an experimental set-up with commercial wireless LAN equipment. We made measurements of the objective audio quality and analyzed the effects of packet losses, both due to real wireless channels and late packet arrivals. Our experiments show that the booster increases the quality of voice best with the hybrid solution and that the performance of Voice over IP can be further improved.

Keywords: Voice over IP, objective speech evaluation, booster, wireless LAN, IEEE802.11, play-out buffer

1. INTRODUCTION

During the last years IEEE802.11 wireless LAN [1] has gained increased popularity and is being deployed, because the price of equipment has dropped and WLAN is easy to install and maintain [2]. WLAN might be a cost saving and local alternative to cellular wireless networks. However, the main usage of wireless LAN has been limited to Internet based services like web, email, and file transfers. Voice transmission over wireless LAN is not considered to the same extent.

The IEEE802.11 specification does define different classes of services, which are essential to support data and voice transmission at the same time. But until recently no commercial product supported both Distributed Coordination Function (DCF) for best effort traffic and the Point Coordination Function (PCF) for prioritized data like voice. Furthermore the PCF mode has not been specified in detail nor its usage has been studied. This is changing [3][4]. We can assume that voice over wireless LAN will become more important and further optimization of its performance should be taken into account - similar to the improvement of TCP over wireless [5].

TCP is showing performance degradations because wireless channels are fluctuating and have high error rates, which are unknown in wireline networks. Worked on the problem "TCP over wireless" [6][7] we wanted to know: Does the voice transmission suffer over wireless links as TCP performance does? And if so, could it be improved by similar mechanisms?

In this paper we will introduce a *Speech Property Based* (SPB) *Booster*, which improves the voice quality over wireless LAN in the case of high packet loss rates. In order to evaluate the performance of the novel booster mechanism, we have set up an experimental measurement environment. The speech audio quality of the normal VoIP architecture has been compared with the speech audio quality using the novel booster mechanism. Because we are interested in the quality of the perceived speech audio, an objective measurement method has been used. The adopted metric is the perceptual distortion (EMBSD) [12], which yields results that have a high correlation with the MOS, the metric used in the case of subjective, human based tests and can hence be used as fast and automatic apprioximation of the more time-consuming subjective tests.

[1] This work has been financed by the European Union's Information Society Technologies Programme (IST) and is part of the MOBIVAS project.

Voice quality is reduced by packet losses even through common voice codecs include concealment algorithms, which interpolate lost audio segments. Packet losses can occur due to errors on the wireless link and late arrivals. We use a simulated play-out buffer to limit the maximal transmission time of voice frames. The play-out buffer drops packets that are too late. It is assumed that sufficient bandwidth is available[2] to avoid losses due to congestion and queuing delays in the backbone.
The basic idea of this booster is based on the observation that not all the packets have the same importance for objective speech quality [9]. Considering the speech signal properties and low bit rate codecs features, it is possible to distinguish between important and unimportant packets [10].
We have developed three algorithms to deal with important packets:

- *Selective packet loss recovery* protects packets by using different configurations of the physical and data link layer and switching between them on a per-packet basis. We altered for our measurements the maximum number of link layer packet retransmissions, which is in compliance with the IEEE 802.11 standard.
- The second solution, called *redundant transmission*, protects the packets classified as important by performing a redundant transmission. The booster mechanism is responsible for duplicating several times the important packets. The Internet protocol stack has no restrictions regarding the duplication of IP packets.
- The third algorithm, called *hybrid solution*, protects the important packets by both adding redundancy and using a selective packet loss recovery procedure.

Our booster is transparent and does not change any protocol specifications. This allows us to use it in existing networks and in parallel with other data transmissions. The algorithm decreases the packet loss rate only slightly, however the voice quality is significantly improved in the case of high error rates. Furthermore the play-out buffer size can be kept low to keep it suitable for interactive telephony.
The next chapter will introduce features of voice transmission. In the third chapter the SPB booster algorithms are presented. Next the measurement set-up and then the results are shown, which are analyzed afterwards. Finally, we conclude.

2. VOICE TRANSMISSION

2.1. Voice over IP

Voice over IP, or IP Telephony, allows us to make telephone calls over the Internet and include signaling, controlling and transport mechanisms [11]. In this paper we focus on the transmission of voice, based on the RTP, UDP and IP protocols. At the application layer, the audio signal is first encoded and a Real Time Protocol (RTP) [12] header is added to the encoded voice frames. The header includes a time stamp, which is later needed for play-out. UDP is used for multiplexing different flows and IP takes care of addressing and delivering the packets to their destination (Figure 1). The corresponding host receives the RTP packets and stores them temporarily to decode and play them in a correct and timely manner.

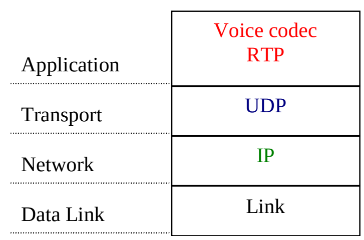

Figure 1 protocol stack

In order to reduce the bit rate of voice transmission audio coding is normally used. Audio encodings compress the raw digital audio stream. With modern codecs it is possible to achieve bit rates of 64kbit/s (G.711), 5.3 kbit/s (G.723.1) or 8 kbit/s (G.729 [13]). These codecs use characteristics of human speech production and perception.
Speech is not a homogenous audio signal and can be distinguished into two different types: voiced and unvoiced. Voiced sounds are created when the human vocal cords vibrate periodically (e.g. vocals). Unvoiced sounds are generated by sending air at a high velocity through a constriction in the vocal tract, generating a kind of turbulence and causing sounds with broad frequency spectrum. In general voiced sounds have a higher energy than unvoiced sounds.

2.2.Voice codec

We limited our studies to one codec, G.729. The reason for our restraint was that the G.729 is well investigated. Rosenberg [14], Sanneck [10] and Long Le [9] used it for similar measurements. Considering one codec we were able to improve the accuracy by conducting more measurements with the same parameter sets. If the measurements are done using other codecs too, they will likely lead to comparable or similar results [15].

[2] The next WLAN generation will have a physical data rate of 54Mbit/s - far more than needed for many voice streams.

The G.729 provides a “toll quality” speech communication, which means a communication as good as the one made with the telephone network. It is a hybrid codec, specified by the International Telecommunications Union (ITU), and employs the CS-ACELP (Conjugate Structure Algebraic Code Excited Linear Prediction) algorithm. The speech is broken into units called frames. In the G.729 these frames are 10 ms long, corresponding to 80 bits. The encoder accepts 16-bit linear PCM data sampled at 8 kHz as input data and produces 8 kbit/s coded data.
This codec provides a good audio quality at a low bit rate. To cope with occasional frame losses the codec includes at the receiver side loss concealment. Concealment algorithms try to appropriately fill gaps caused by losses. This can be done for example by repeating the previous sounds or by interpolation. The applicability of the G.729 concealment is limited to isolated losses. As soon as consecutive frame losses occur, the audio quality decreases quickly.
Rosenberg has measured the resynchronization time of the G.729 decoder after the loss of multiple consecutive frame losses [14]. Resynchronization is achieved if the difference between the decoded audio streams (with losses and without) falls below a certain threshold, i.e. until good and bad segments sound similar again. Sanneck and Long Le extended these measurements by altering the position of the frame losses, especially at unvoiced and voiced transmissions (fFigure 2). Through their results we can see how the resynchronization time of the decoder depends strongly on the position of the lost frames.

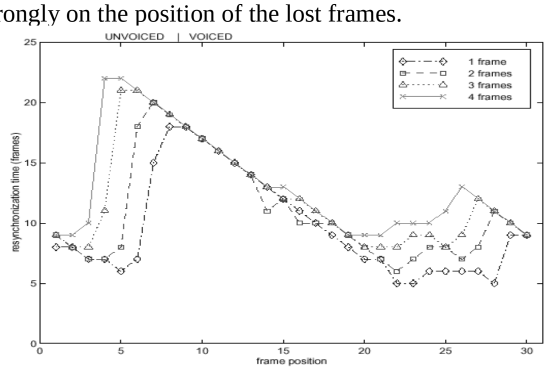

Figure 2 – G.729 resynchronization time [10]

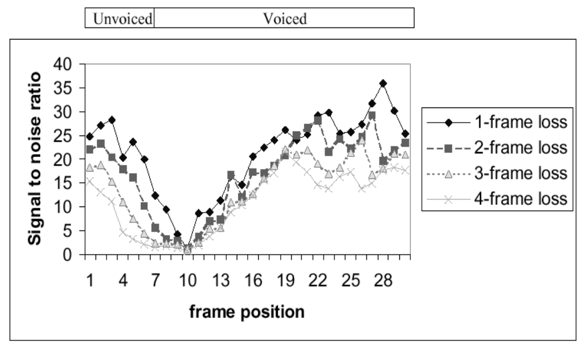

Figure 3 – SNR of the G.729´s decoded speech signal after the loss of a number of frames [10]

Also, they have then conducted the same experiments measuring the accumulated signal to noise ratio (SNR) after a frame loss. The SNR has been calculated over the next 15 consecutive frames after loss. The consecutive SNR at an unvoiced/voiced speech transition is shown in figure 3 with respect to the position of the loss. They also obtain similar results by using objective voice measurements tools. These observations have been made only for G.729, but in the meantime Sun and Ifeachor confirm them for other codecs too [15]. We can conclude that losses have different impact on the reconstructed speech signal, depending on the position at which they occur.
The importance of the speech frames transmitted on the network can be ranked:

- The loss of a speech frame after an unvoiced-voiced transition is the worst case. The reason is that the decoder tries to conceal the loss using the filter parameters and the excitation of the previous frame, which was an unvoiced frame. Moreover, the voiced sounds have quasi-periodic characteristics highly different from unvoiced sounds, so they cannot be easily predicted.
- The loss of a voiced speech frame causes a resynchronization time that is not short, but not so long as in the previous case.
- The loss of unvoiced frames seems to have little impact on the speech quality.

2.3. Voice quality evaluation

The quality of an audio signal, which has been subjected to any kind of processing, can be determined by the normal listener evaluating the processed signal. The impressions of listener are subjective and may not be correlated with technical measurements. For example, the signal to noise ratio is not taking into account the characteristics of the human auditory system and is only a rough estimate of the evaluation of a human being. Objective measurement methods instead implement a model of the human audio perception. They provide perceptual measurement of the speech audio quality and therefore a technical measure for a perceptible quantity. Modern objective methods on speech audio signals produce results which are highly correlated with the mean option score (MOS), the metric used for subjective tests. Substantial tests have been made to compare subjective and objective results [16], and they acknowledged the correlation of the objective and subjective evaluations (in this cases using “TOSQA”). Recently the Perceptual Evaluation of Speech Quality (PESQ) algorithm has been officially approved as a new ITU-T recommendation P.862. PESQ was developed to be used on end-to-end voice quality testing under real network conditions. We can conclude that the accuracy of objective evaluation is high enough so that it is possible to limit the number of time and money consuming subjective tests – at least for the development and evaluation of telecommunication systems like VoIP.

For our measurements we applied the Enhanced Modified Bark Spectral Distortion (EMBSD) metric. It is based on the bark spectral distortion (BSD) measure, which is an objective measure that emulates several of only perceptually significant auditory attributes. In the Enhanced Modified Spectral Distortion (EMBSD) [8] new metrics and additional effects are taken into account. The EMBSD calculates distortion values using the original audio sample and the processed speech. A value of zero means no distortion and a good speech quality. G.729 has a minimal distortion of about 0.9 due to coding losses. Higher values indicate a worse quality or a lower MOS value.

The EMBSD measurement tool is available at no costs. It has been compared with the PESQ [17] and can predict the speech quality in various conditions for a given coder, but not across different codecs. We are aware that results of objective measurement tools have to be used with care and do have an error margin, but we are thankful, that we could use EMBSD.

3. THE SPB BOOSTER MECHANISM

As mentioned previously, not all packets have the same importance in a speech audio stream. Consequently, a reasonable hypothesis is to classify the packets that are going to be transmitted and to protect those packets which are classified as important. This is the basic idea underlying this work.

To implement our mechanism we applied a suitable design paradigm: Protocol boosters are by definition software and/or hardware modules that transparently enhance existing protocols [18]. Some guiding principles of the boosters are:

- No modification of the syntax or semantics of the exchanged end-to-end protocol messages.
- Transparency with regard to the protocol being boosted.
- The possibility of adding, deleting, queuing or delaying end-to-end protocol messages.
- Elimination of any actions of the booster which do interrupt the end-to-end communications.
- Instantiation or revocation of on-the-fly based policies, such as observation of the network behavior, packet source and destination, time of day, or composition rules with other boosters.

In this implementation the booster mechanism enhances the perceived Quality of Service provided at the user level. This is obtained by considering the importance of the packets within the speech audio stream. As a consequence, it is possible to reduce the impact of the losses on the quality of the speech by “protecting” those packets that are selected as ‘important’.

In figure 4 the architecture of the booster mechanism is shown in the case of a sending mobile terminal. It is composed of a transmitting side, located on the mobile terminal, and a receiving side, located on the bridge. Both parts of the booster reside at the MAC- and link layer. The part located on the mobile terminal is responsible both for classifying the packets (the “analysis” block) and for providing this information to the booster. Consequently, the MAC/link layer adopts or not adopts a protected transition, depending on the classification of the packet. The part located in the bridge is responsible for making the booster mechanism transparent to the rest of the system.

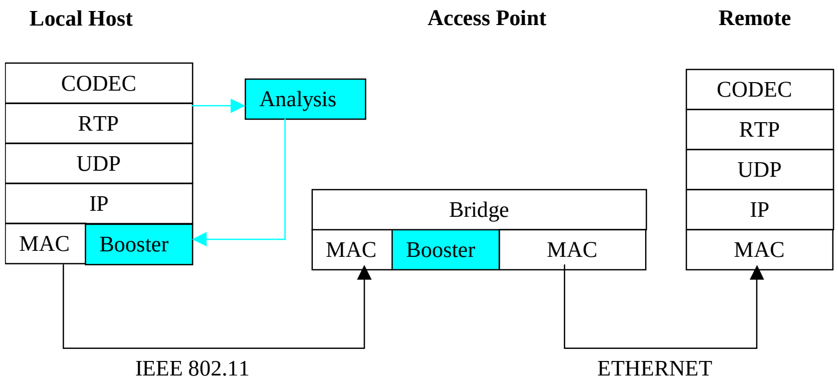

Figure 4 – speech property booster

3.1. Which packets are to be protected?

The mechanism, which we use to classify the packets as voiced or unvoiced, is included in the ITU implementation of the G.729 decoder. The encoded frames are decoded to analyze them. This consumes some processing resources on the mobile terminal but for our measurements we did not try to improve the execution time of the classifier. It should be possible to include the detection in the encoder or use parameters in the frame structure and to limit thereby the processing overhead.

The audio quality is most influenced by losses within the first 15 speech frames after an unvoiced/voiced transition (Figure 2). Therefore we mark these frames as important, as long as they are voiced frames. Sanneck has run simulations and it seems that 10 to 20 are appropriate values for the number of frames to be protected depending on general network conditions. In total about one fourth of all speech packets are marked as protected in our 15s speech audio sample.

3.2. How to protect the important packets against loss?

The loss probability of the important packets has to be reduced. But how to decrease the loss rate most efficiently? It can be done on several layers. In this paper we will not discuss end-to-end transport layer error recovery techniques based on forward error correction (FEC) and automatic repeat request (ARQ) [19] as we assume that most losses occur at the wireless link. Of course, in the case of network congestion these transport layer mechanisms have to be taken into account or a dynamic rate adaptation of the coder has to be applied.

For protecting the packets classified as important we have chosen three solutions, which are placed in the link layer: The *selective packet loss recovery*, *redundant transmission*, and the *hybrid solution* are described next.

3.3. Selective packet loss recovery

For ensuring a higher reliability in the transmission of packets from the mobile station to the bridge, several parameters of the wireless card configuration can be set, in compliance with the IEEE 802.11 standard. However, the real possibility of changing the value of the configuration parameters of link layer depends on the features of the particular wireless LAN card. We have worked with three different commercial wireless LAN cards: Aironet, Lucent WaveLan, and Intersil PRISM2 cards. The later is most advanced, because it has a higher level of programmability and it is well documented. We studied the use of the following parameters, which can be altered in the PRISM2 card and partly in the others, too.

Using *Transmit Power* values higher then the default implies that the driver of the wireless card will use a higher power to transmit the packets, so the probability that a packet is lost will decrease. But often the TX power is already set to the maximum value.

It is possible to protect the important packets by dynamically changing the *data rate* and *modulation*. In fact, when transmitting at a lower data rate, the packet loss probability decreases. Nevertheless, the IEEE 802.11b WLANs standard already implements a dynamic rate shifting in order to adjust automatically the data rate to compensate the changing radio channel conditions. In our measurement we used this automatic rate adaptation.

The IEEE802.11 wireless LAN retransmits erroneous packets automatically. The receiving side acknowledges each transmitted packet immediately. If the transmitter does not receive an acknowledgment, it assumes that the packet has

been lost and restarts the transmission after some time until a maximal retransmission limit has been reached. Normally a packet is transmitted at most about ten times, but this depends on the packet size or the IEEE802.11 implementation. The IEEE 802.11 standard does not define retry limit. In this work, it has been chosen to protect the important packets using a higher retry limit.
Using the Intersil PRISM2 wireless card it was possible to change the Linux device driver and to implement the change of the number of retransmissions on a per-packet basis. Intersil has included this feature in the firmware of the medium access controller. (The MAC documentation is available under NDA.) Hence, we were able to extend the device driver to support the per-packet maximal retry limit.
There are some reasons to limit the range of the retry limit. A G.729 packet is transmitted at a data rate 11Mb/s within 571µs. Therefore, it should be possible to transmit 35 packets within 20ms, which 20ms is the time between two consecutive G.729 packets. But the DCF mode of IEEE802.11 includes a backoff algorithm, which delays the transmission of packets. This delay is a random number between 0 and the contention window. The size of the contention window ranges from minimal 7 slots and maximal 255 slots (DSSS slots are 20µs long) and increases exponentially after each unsuccessful transmission attempt. The backoff is intended to avoid collisions of multiple transmitters that try to send at the same time, but it works in the same way on a channel with a high packet error rate. The table shows the mean transfer time of a packet for 11Mbit/s in the absence of any other traffic and senders. The equation can be found in [20].

No of Retries	Mean Contention Window in µs	Mean Total Transfer Time in µs
0	0	571
1	70	1212
2	150	1933
2	310	2814
3	630	4015
4	1270	5856
5	2550	8977
6	2550	12098
7	2550	15219
8	2550	18340
9	2550	21461
35	2550	99486

Table 1 DCF mode: mean transfer time

In the IEEE802.11 PCF mode a backoff algorithm is not needed, because the base station schedules the medium access. Therefore, a scheduler can force the retries. The scheduling algorithm is not yet defined for IEEE802.11.

Next the selective packet loss recovery algorithm is described. The following source code is been run for each packet to transmit. First the function `classify` is called to classify the packet as important, which means that within the last 14 frames an unvoiced/voiced transition occurred. For important packets the link layer protection is switched on by setting the retry limit to 35 and the packet is sent. Otherwise, the packet is sent without any protection at a retry limit of 10, the default value.

```
if(classify(datagram_packet))
    card.set_protection();
else
    card.no_protection();
send(datagram_packet);
```

Of course an additional delay is potentially introduced for the transmission of the protected packets, while going over the wireless link. In fact, when using a higher retry limit in case of bad channel conditions the packets will need a larger amount of time to reach the remote host, due to the higher number of link layer retransmissions. Otherwise, without the former mechanism, they could be lost. The receiving side of the booster does not have any function in this case. The booster mechanism is in this case already transparent to the rest of the network.

3.4. The redundant transmission

A second possibility consists in protecting the packets classified as important by adding Forward Error Correction (FEC) like redundancy to the transmission. Using FEC is a well-known mechanism to protect speech transmission for both cellular wireless networks and interactive multimedia on the Internet. In this case the important packets are protected using packet duplication. Each important packet is transmitted several times.
Common Internet application and transport protocols have been designed to tolerate multiple copies of the same packet. IP does not provide any mechanism to filter out these packets. Duplicated packets can occur in the case of network failure or misuse. The network control and management might interpret extensive packet duplications as aggressive act and therefore “tolerance” of transport protocols should not be used for FEC. One other issue is that it causes additional traffic in the backbone, where duplicated packets are not needed. Our booster mechanism filters out duplicated packets after the wireless link.

One reason for using redundant transmissions is that the wireless channel is highly time dependant. A wireless channel may be considered stationary only during very short periods. Through several measurements in an indoor environment it is possible to observe the high variability of the channel.
During the time between two consecutive packets the important packets may be sent several times in order to find better channel conditions and to reduce the loss probability for these packets. On the receiver side the booster mechanism will discard the duplicated packets. However, also in this case it is not possible to transmit the same packet too many times. Each retransmission requires a certain amount of time. Therefore, if the packet reaches the remote side with the n-th transmission due to bad channel conditions, a delay may be added that is too in regard to the real-time requirements. In this work the important packets have been sent three times.
Due to the practical difficulties in modifying the driver of the wireless card and the software of the base station, we implemented the redundant transmission at the application layer in order to test the performance of this solution. The retransmissions are performed at the application layer and the elimination of the duplicated packets is performed directly in the remote host, at the application layer too.
The algorithm that implements the protection of the important packets is:

```
if(classify(datagram_packet))
{
    send(datagram_packet);
    Thread.sleep(1);
    send(datagram_packet);
    Thread.sleep(1);
}
send(datagram_packet);
```

Using this kind of retransmissions some redundancy is added because the important information is transmitted three times. The throughput needed to run this algorithm is higher compared to the normal way of transmission. However, on the receiver side the duplicated packets are discarded and the throughput needed from the base station to the remote host through the wired link is again 8 Kbit/s, the original bit-rate of the codec.

3.5. The hybrid solution

At last the hybrid solution is adopted. It includes both selective packet loss recovery and redundant transmission:

```
if (classify(datagram_packet))
{
    card.set_protection();
    send(datagram_packet);
    Thread.sleep(1);
    send(datagram_packet);
    Thread.sleep(1);
}
else
    card.no_protection();
send(datagram_packet);
```

4. MEASUREMENT SET-UP

4.1. Measurement environment

In this paper a scenario is considered which consists of a mobile host in an IEEE 802.11 wireless local area network. The mobile host transmits speech packets through the wireless link to the base station that is connected to an Ethernet network. Through the base station and the Ethernet the speech packets reach the remote host.

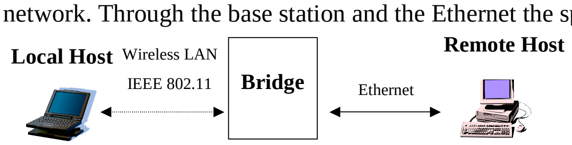

Figure 5 - Hardware of measurement set-up

The measurements took place at our university office building. The building has walls out of Ferro-concrete, steel doors, and contains numerous electrical power installation and workstations [21]. Because of the growing use of the 2.4 Gigahertz frequency band one of us conducted the test in the evening and at night times to

assure that no other sources are transmitting in the same band.
A notebook with a 400MHz Pentium II, Linux 2.2 and an Intersil PRISM2 wireless PC Card [22] using the linux-wlan driver 0.1.7 [23] has been used as transmitter. Apple's AirPort was the base station and has been connected to an Ethernet LAN. The receiving host has been a PC with a Celeron 700MHz CPU, connected to the LAN, too. Others have used the LAN only negligibly during the measurements.
The wireless LAN equipment did not support the PCF mode; therefore we had to use DCF. We made measurements with background traffic, too, however the results were unreliable, being influenced largely by the burstiness of TCP traffic. Therefore we switched off all other senders. With the usage of PCF the influence of bursty data traffic should be minimal.

4.2. Simulated Play-out Buffer

In the Internet packets are transmitted with no time guaranties and arrive with a different transmission delay at the receiving host. This variation of delay causes delay jitter. However, the G.729 audio frames are generated at a fixed rate of 10ms and the receiver plays out the frame at the same rate. To avoid the effect of delay jitter a play-out buffer is used. The play-out buffer stores voice packets for a certain time until they are decoded. The size of the play-out buffer is selected in such a way that most packets are played on time. Play-out buffer in multimedia terminals have the drawback that they do not know the transmission delay at the beginning, causing initial inaccuracy, until the channel conditions are known better. During transmission they are adapted to the current delay conditions [24].
In our measurement the play-out buffer differs from common implementations, e.g. [25]. During the measurement the packet arrival time is stored. After the complete audio sample has been transmitted, the time stamps are used to calculate the effect of the play-out buffer. To get the lower bound of the play-out buffer for each packet the relative transmission delay is calculated. The packets are generated at a regular interval at the sender (figure 6, line t0). They are transmitted and arrived at the receiver with a delay. It is possible to identify the fastest packet, even without knowing its absolute delay. The fastest packet is used to set the optimal lower bound of the play-out buffer (figure 6, line t1). Using a given buffer size the play-out can determine all packets that are too late (or which are right of line t2). The novel simulated play-out buffer has the advantage that the effect of different play-out buffer sizes can be analyzed for the same measurement by changing the upper bound of the play-out buffer.

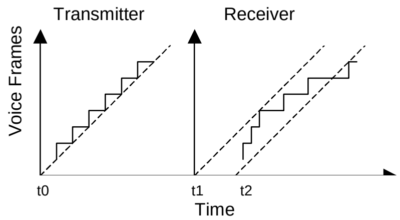

Figure 6 – Simulated Play-Out Buffer

4.3. Speech Samples

The speech sample, which has been used for taking the measurements, consists of about 15 seconds of male and female English speech. In total 1125 encoded speech frames are transmitted through the wireless link to the remote host. Since two speech frames are encapsulated in one RTP packet a total of 564 packets are sent through the wireless link. Each speech text has been transmitted 100 times, after which the position of the transmitter has been altered. On the receiving side we collect whether and when the packets have arrived. The results are simultaneous analyzed by the simulated play-out buffer and then by a speech quality evaluation.

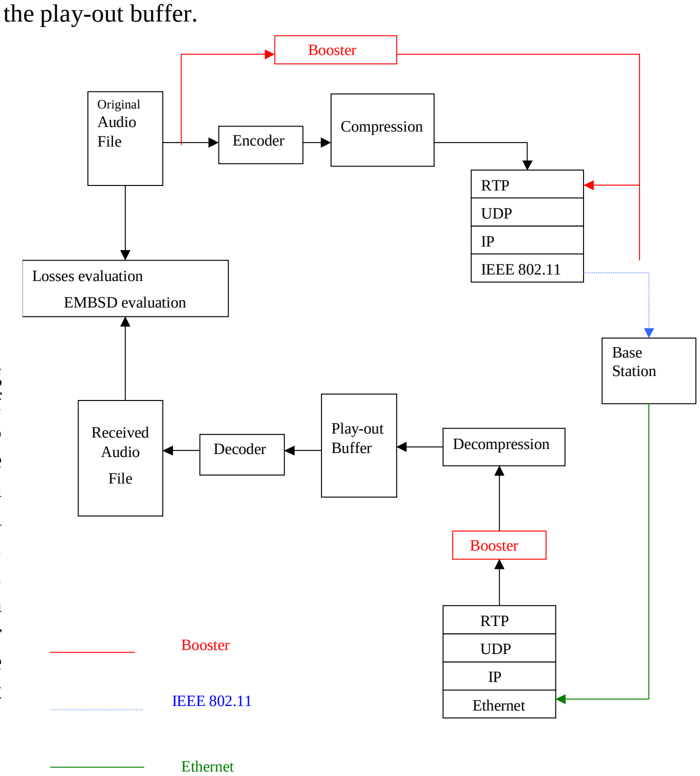

Figure 7 - complete measurement set-up

4.4. Overall measurement set-up

To summarize, the complete measurement set-up is displayed in figure 7. The original speech sample is first encoded and then analyzed for voiced/unvoiced transitions. Two frames are combined in one packet. This packet is transmitted using RTP/UDP/IP and wireless LAN. If the packet is classified as important the booster controls the transmission. The base station receives the packet and forwards it to the corresponding host, where the booster filters out duplicated packets. The packet arrival time is stored. After measurement the play-out buffers are simulated using the time stamps of the packets. For multiple play-out buffer sizes the packet loss rate is calculated. The remaining frames are decoded. The resulting audio sample is being compared with the original using EMBSD and so the voice quality is evaluated.

5. MEASUREMENT RESULTS

The following measurement results present the cases of high error rate, which occur on the outer limit of the transmission range or during movement of the mobile terminal. We skipped the measurements which have few or no packet losses, because the voice quality does not show any degradation. Most of the time we noticed a good wireless channel or no connection could be established. Cases of high error rates were seldom. The following plots are based on 400 measurement, each 15s long.

The four analyzed algorithms were:

- Normal VoIP architecture (*unprotected*)
- *Selective packet loss recovery* booster
- *Redundant transmission* booster
- *Hybrid* booster

Figure 8 has on the y-axis the loss rate and on the x-axis the play-out buffer length. The reported loss rate is the speech frame loss rate. Considering that two speech frames are encapsulated in one IEEE 802.11 packet, the channel packet loss rate is half the frame loss rate. In the figure the packet loss rate is between 2% and 10%. The play-out buffer size is measured in frames. Therefore, the scale has to be multiplied by 10ms to get the time.

In figure 9, the perceptual distortion (EMBSD) as a function of the play-out buffer size is plotted for all the analyzed cases. Lower values are corresponding to a better speech quality. Identical audio sample a distortion of zero. Due to the coding losses of G.729 the values between the original and the decoded/encoded sample is 0.9.

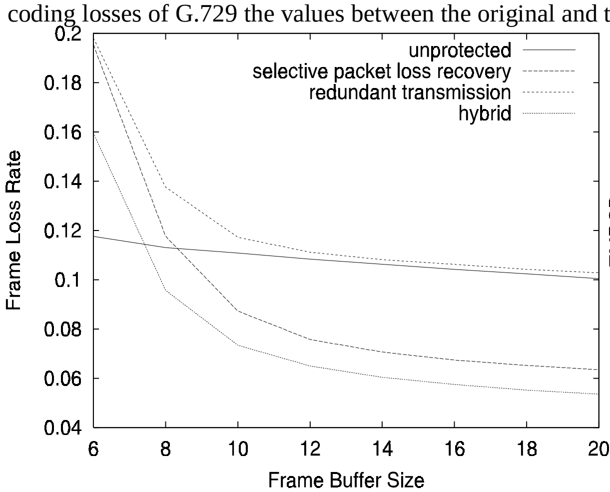

Figure 8 – losses/buffer size

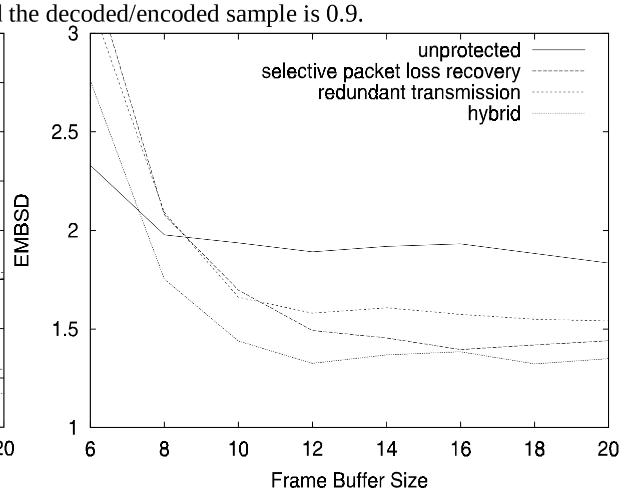

Figure 9 – EMBSD/buffer size

In Figure 10 the variance of the perceptual distortion as a function of the play-out buffer size is plotted. Figure 11 is combined out of figure 8 and 9. It compares the frame losses rates with the EMBSD value.

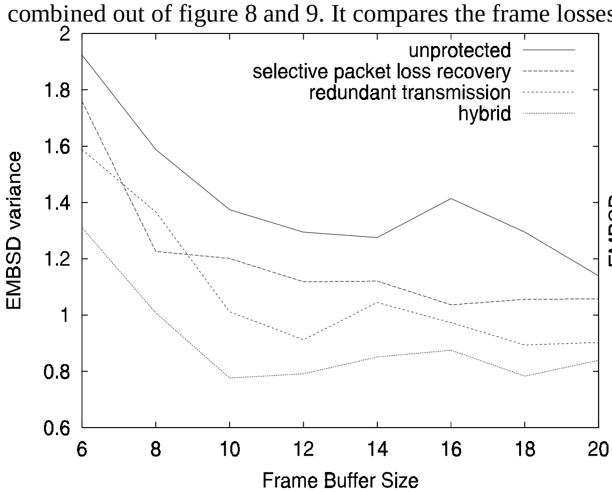

Figure 10 - EMBSD variance/buffer size

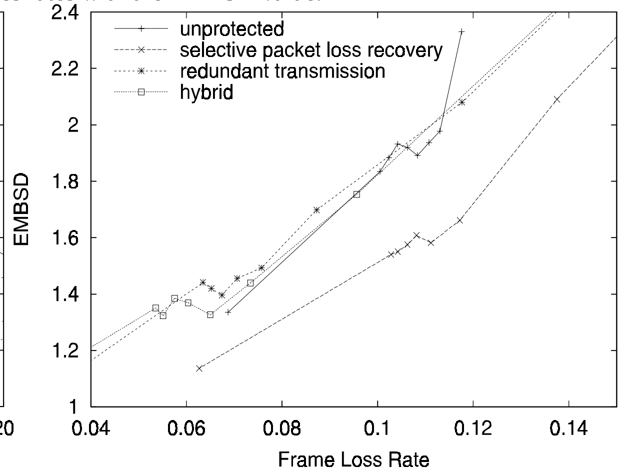

Figure 11 – EMBSD / losses

6. ANALYSIS

Starting with figure 8 we first analyze the influence of the size of the play-out buffer. For all three booster algorithms the loss rate decreases for values up to 100ms. For values above 100ms the loss rate is more stable and decreases only slightly. The time of one packet transmission on the wireless link is in all cases between 571µs and 100ms. Therefore we can conclude that the loss rate is not only influenced by the wireless link but also by other sources. Indeed, in the current measurement set-up the time stamps, which are used for the play-out buffer simulation, are taken at the application layer. This might cause additional delay variations due to process scheduling, garbage collection of the Java virtual machine and queuing in the network. Furthermore, an additional processing overhead of the booster algorithms can be identified compared to the unprotected case. The reason for measuring at the application layer is that for the perceptual quality the mouth-to-ear delay is most important. The application layer is a close approximation to it because only the acoustical processing and propagation is missing. However, the application layer processing has to be optimized and the reasons of the delay variation have to be studied further. It might the useful to place the telephony application in a real-time execution environment to avoid additional delay variations.

In the following we will concentrate on play-out buffer sizes between 100ms and 200ms, which are values that are still acceptable for interactive audio communication. In figure 8 the loss rate of the four algorithms can be compared. Whereas the *selective packet loss recovery* has no influence on the loss rate - it even increases it – the *redundant transmission* and the *hybrid* cases decreased the losses by about 30% and 40%. Comparing it with figure 9 we can see that in it the case of *selective packet loss recovery*, even through the loss rate is similar to the *unprotected* case, the perceived speech audio quality is improved. Both *redundant transmission* and the *hybrid* cases improve the voice quality too. But what are the reasons for these improvements?

In Figure 10, the variance of the perceptual distortion as a function of the play-out buffer size is plotted. The gain achievable with the new booster mechanism is evident. The variance of the *hybrid* case is decreased by about 40% compared to the unprotected transmissions. The other cases are in between.

Figure 11 compares the packet loss rate with the EMBSD perceptual distortion for three cases. We can see roughly two lines: The first line is based on data from the *unprotected, redundant transmission,* and *hybrid cases*. The second consists of data from the *selective packet loss recovery* case. The second has a far better correlation of losses versus EMBSD. The losses have less negative influence on the voice quality. This might be due to the fact that if losses occur, they are likely due to unimportant packets and therefore the perceived speech audio quality is kept at a good level.

7. RELATED WORK

For Voice over IP Bolot [26] has proposed to use an open loop error control mechanism based on forward error correction. He argued that ARQ mechanisms are not acceptable for interactive audio applications because they dramatically increase end-to-end latency. The discussed solutions are only end-to-end.

Bakin [2] has developed an FEC Booster for UDP applications over terrestrial and satellite wireless networks. The booster is similar to our redundant solution, but protects multiple consecutive packets by additional parity packets. Therefore, it increased the latency, because the lost packets can only be reconstructed after receiving all other packet in a FEC block.

8. SUMMARY

This work has started to analyze potential quality improvements of voice over wireless LAN. The experimental results show that it is possible to increase the voice quality in the case of high packet error rates. The main research contributions are the development of new booster mechanisms, which improve the perceived audio quality, and a set-up of an experimental measurement environment for testing the end-to-end performance.

The booster mechanism selectively protects packets, which are classified as important for the perception of the voice. The SPB booster adopts a speech aware transmission policy for VoIP packets through the wireless link. The packets marked as relevant to the perception of the voice are sent in a "protected" way. Three solutions have been proposed, *selective packet loss recovery*, *redundant transmission, and hybrid solution*. We implemented the SPB booster mechanisms for a Linux operating system and IEEE 802.11 wireless LAN drivers and used this system for the measurements.

The results of our measurements have been analyzed regarding the packet loss rate, the speech quality and the play-out buffer size. The SPB booster leads in our experiments to an improvement of the audio quality. The voice quality increases from the *unprotected case*, the *selective packet loss recovery, redundant* to the *hybrid* case. Interpreting the measurement results we can say that the *hybrid* and *redundant* transmission causes these gains by a reduced loss rate and in the case of *selective packet loss recovery* the voice quality is improved by an intelligent distribution of the packet losses, which has a smaller negative effect on the speech quality. The measurement results are preliminary and further studies have to be made to improved the statically accuracy and identify further reasons of delay jitter and loss.

During the development of the novel booster mechanism several problems have been encountered. It has been quite difficult to deal with the device drivers to change the link layer properties because of a lack of documentation. Real drivers introduce unknown side effects that make the interpretation of the experimental results hard. The manufactures should open the documentation of the wireless networking equipment to allow the development of application-specified improvements.

We are continuing our initial experiments because further research regarding Speech Property Based (SPB) boosters promises to produce encouraging results. This work has been successful in showing that the perceptual voice quality over wireless LAN can be improved - even using an experimental set-up.

ACKNOWLEDGEMENTS

We would like to express our gratitude to Ing. Enrico Gregori for having made the visit of Iacopo Carreras at Technische Universität Berlin possible.

REFERENCES

1. IEEE Computer Society LAN MAN Standards Committee. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std 802.11-1997. The Institute of Electrical and Electronics Engineers, New York, New York, (1997).
2. Wireless LAN Association, "Wireless local area networking: ROI / cost-benefit study", available via WWW at URL: http://www.wlana.com.
3. M. Veeraraghavan, N. Cocker, and T. Moors, "Support of voice services in IEEE 802.11 wireless LANs", *Proc. Infocom*, Anchorage, Alaska, (April 2001).

4. A. Köpsel, J.-P. Ebert, and A. Wolisz, “A performance comparison of point and distributed coordination function of an IEEE 802.11 WLAN in the presence of real-time requirements,” in *Proc. 7th Intl. Workshop on Mobile Multimedia Communications (MoMuC2000)*, Tokio, Japan, (October 2000).
5. H. Balakrishnan, V.N. Padmanabhan, S. Seshan, and R.H. Katz. "A comparison of mechanisms for improving TCP performance over wireless links," `96 Conference, pp. 256-269, Stanford, California, (August 1996).
6. M. Schläger, B. Rathke, S. Bodenstein, and A. Wolisz, “Advocating a remote socket architecture for internet access using wireless LANs,” *Mobile Networks,* Special Issue on Wireless Internet and Intranet Access, *Kluwer Academic Publishers,* 6(1):23-42, (January 2001).
7. F. Fitzek, B. Rathke, M. Schläger, and A. Wolisz, “Simultaneous MAC packet transmission in integrated broadband mobile system for TCP,” Proc. of ACTS SUMMIT 1998, pp. 580--586, Rhodos, Greece, (June 1998).
8. W. Yang, M. Benbouchta, and R. Yantorno. “Performance of the modified spectral distortion as an objective speech quality measure,” in Proc. *ICCASP*, vol. 1, Seattle, Washingtoon, (Mai 1998).
9. N.Long Le, “Development of a loss-resilient internet speech transmission method,” *Diploma Thesis, GMD Fokus /Lehrstuhl für Telekommunicationsnetze, TU Berlin,* Berlin, Germany, (Mai 1999).
10. H. Sanneck, “Packet loss recovery and control for voice transmission over the internet,” *PhD Thesis, GMD Fokus*, Berlin, Germany, (Oktober 2000).
11. H. Schulzrinne, and J. Rosenberg, "Internet telephony: architecture and protocols - an IETF perspective," *Computer Networks and ISDN Systems,* Vol. 31, pp. 237-255, (February 1999).
12. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: a transport protocol for real-time applications,” RFC 1889, Internet Engineering Task Force, (January 1996).
13. International Telecommunication Union. “Coding of speech at 8 Kbit using conjugate-structure algebraic-code-excited linear-prediction,” ITU-T Recommendation G.729, (March 1996).
14. J. Roseberg, “G.729 error recovery for internet telephony,” *Project report, Columbia University*, (1997).
15. L. Sun, G. Wade, B. Lines, and E. Ifeachor, “Impact of packet loss location on perceived speech quality,” *2nd IP-Telephony Workshop*, pp. 114-122, New York, USA, (April 2001).
16. J. Berger, H. Klaus, I. Kliche, T. Scheerbarth, H.-W. Gierlich, and F. Kettler, ”Report of 1st ETSI VoIP speech quality test event,” *ETSI EP TIPHON 22, Temp. Doc. 031*, Bethesda, USA, (March 2001).
17. C.S. Chow, “Research on objective speech quality measures”, *Master Thesis, Massachusetts Institute of Technology, USA*, (February 2001).
18. D.C. Feldmeier, A.J. McAuley, J.M. Smith, D.S. Bakin, W.S. Marcus, and T.M. Raleigh, "Protocol boosters," *IEEE Journal on Selected Areas in Communications,* Special Issue on Protocol Architectures for 21st Century Applications, 16(3):437-444, (April 1998).
19. G. Carle, and E.W. Biersack, “Survey of error recovery techniques for IP-based audio-visual multicast applications”, IEEE Network Magazine, 11(6):24-36, (November 1997).
20. F. Cali, M. Conti, and E. Gregori, "IEEE 802.11 wireless LAN: capacity analysis and protocol enhancement," *Proc. IEEE INFOCOM*, Piscataway, New Jersey, (1998).
21. B. Rathke, M. Schlager, and A. Wolisz, "Systematic measurement of TCP performance over wireless LANs," *Technical Report TKN-01BR98, Telecommunication Networks Group, Technical University Berlin*, Berlin, Germany, (December 1998).
22. Intersils PRISM2 Reference Design, available via WWW at URL: http://www.intersil.com/design/prism/
23. linux-wlan(tm) Project, available via WWW at URL: http://www.linux-wlan.com
24. R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne, "Adaptive playout Mechanisms for Packetized Audio Applications in Wide Area Networks," *In Proceedings of the Conference on Computer Communications (IEEE Infocom),* pages 680--688, Toronto, Canada, (June 1994).
25. H. Schulzrinne, “NEVOT Implementation and Program Structure,“ *University of Massachusetts, Technical Report*, (1996).
26. J.-C. Bolot, and A. Vega-Garcia, "The case for FEC-based error control for packet audio in the internet," *ACM Multimedia Systems,* (1997).
27. D. Bakin, W. Marcus, A. McAuley, and T. Raleigh. “An FEC booster for UDP application over terrestrial and satellite wireless networks,” *International Mobile Satellite Conference (IMSC 97),* Pasadena, California (June 1997).