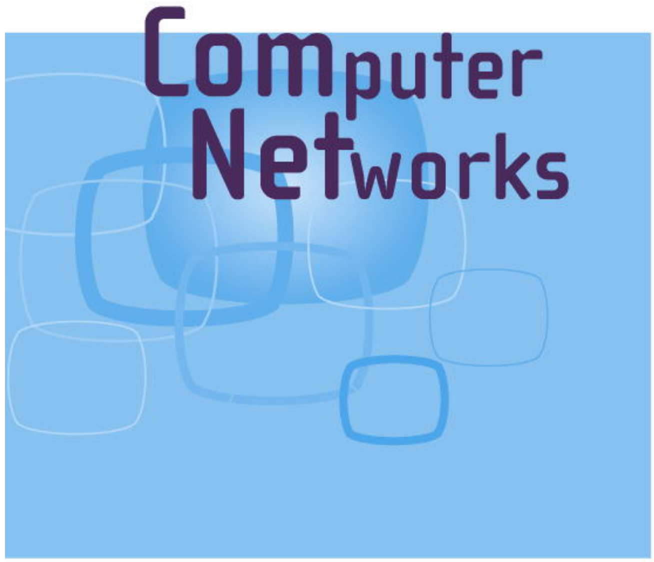
Volume 52 Issue 3 22 February 2008 ISSN 1389-1286



Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Computer Networks 52 (2008) 650–666
Computer Networks
www.elsevier.com/locate/comnet

Analysis of Skype VoIP traffic in UMTS: End-to-end QoS and QoE measurements

Tobias Hoßfeld *, Andreas Binzenhöfer

University of Würzburg, Institute of Computer Science, Department of Distributed Systems, Am Hubland, 97074 Würzburg, Germany

Received 28 January 2007; received in revised form 1 August 2007; accepted 9 October 2007
Available online 6 November 2007

Responsible Editor: Ing. W. Kellerer

Abstract

In the future Internet, multi-network services will follow a new paradigm in which the intelligence of the network control is gradually moved to the edge of the network. This impacts both the objective Quality of Service (QoS) of the end-to-end connection as well as the subjective Quality of Experience (QoE) as perceived by the end user. Skype already offers such a multi-network Voice-over-IP (VoIP) telephony service today. Due to its ease of use and a high sound quality, it becomes increasingly popular in the wired Internet.

UMTS operators promise to offer large data rates which should suffice to support VoIP calls in a mobile environment. However, the success of those applications strongly depends on the corresponding QoE. In this work, we analyze the theoretically achievable as well as the actually achieved quality of IP-based voice calls using Skype. This is done performing measurements in both a real UMTS network and a testbed environment. The latter is used to emulate rate control mechanisms and changing system conditions of UMTS networks. The results show in how far Skype over UMTS is able to keep pace with existing mobile telephony systems and how it reacts to different network characteristics. The investigated performance measures comprise the QoE in terms of the MOS value and the QoS in terms of network-based factors like throughput, packet interarrival times, or packet loss.

Keywords: Skype VoIP; Edge-based intelligence; 3G UMTS networks; End-to-end QoS; Quality of experience QoE; Measurement

1. Introduction

The future generation of telecommunication systems will be a mixture of heterogeneous networks which comprise various access technologies. A mobile user, e.g., will move through a landscape of wireless technologies like WLAN, UMTS, or WiMAX. This leads to multi-network services which must work transparently to the underlying network infrastructure and independently of the user's current location and access technology. In the range of VoIP, this implies that an arbitrary service subscriber must be able to communicate with any other voice user. A UMTS subscriber might,

* Corresponding author. Tel.: +49 931 8886642; fax: +49 931 8886632.
E-mail addresses: hossfeld@informatik.uni-wuerzburg.de (T. Hoßfeld), binzenhoefer@informatik.uni-wuerzburg.de (A. Binzenhöfer).

1389-1286/$ - see front matter
doi:10.1016/j.comnet.2007.10.008

e.g., use IP telephony to talk to a wireline user of the plain old telephone system (POTS). In this sense, a multi-network service can be regarded as a logical overlay on top of different access networks.

As a result new applications and services arise in the Internet which implement the control of network traffic on application layer. Thus, the intelligence is moved away from the core network towards the edge of the network. Popular examples for such edge-based services are already existing Peer-to-Peer (P2P) file-sharing networks, like eDonkey or BitTorrent, or the Skype Voice-over-IP (VoIP) client. Skype is a proprietary application which is based on P2P technology. It offers rapid access to a large base of users, seamless service operation across different types of networks (wireline and wireless) with an acceptable voice quality [1], as well as a distributed and cost-efficient operation of a new service. The good voice quality of the Skype service is achieved by appropriate voice codecs, such as iSAC and iLBC [2], as well as by adapting the traffic rate of the sender to the current conditions in the network which are described by classic *end-to-end quality of service (QoS)* parameters, like packet loss or jitter.

However, the end-to-end *quality of experience (QoE)* perceived by the user will be the essential criterion for the subscriber of a service. The QoE is a subjective measure describing the user's perception of the overall value of the provided service. To be more precise, a voice user is not interested in performance measures like packet loss probability or received throughput, but mainly in the current quality of the speech call. The VoIP user expects a reliable voice communication in a good quality and the availability of the service itself. For assessing the quality of voice calls, the *mean opinion score (MOS)* [3] is an abstract form of expressing the QoE.

The aim of this work is to investigate if mobile Skype is feasible in current 3G networks. We performed measurement studies in a public UMTS network as well as in an emulated UMTS environment. The emulation is used to push the network to its limits in order to investigate how Skype reacts to such conditions in the network. The benefit of Skype's edge-based intelligence will also be discussed analytically by studying the replication of voice datagrams. Additionally, we will characterize the traffic profile for common QoS parameters and measure the QoE in terms of MOS values.

The remainder of this paper is organized as follows. Section 2 gives an overview of existing work on the Skype application. The performance objectives of the end-to-end connection are defined in Section 3, where we also demonstrate the experimental setup used to measure the QoS and the QoE. Section 4 considers rate-controlled dedicated channels in UMTS which use a fixed bandwidth for a certain time. The impact of the emulation tools and the different possibilities to emulate network characteristics are discussed in Section 5. The results indicate that the replication of voice datagrams can overcome random packet drops. The general benefit of replication will also be described analytically. After that, our measurements in a public UMTS network are analyzed to study the impact of the network infrastructure on the voice quality in Section 6. The QoE adaptation of Skype and its reaction to network changes are addressed in Section 7, which demonstrates the variety of possibilities to enable edge-based intelligence. Finally, Section 8 concludes this work.

2. Related work

The immense success and popularity of Skype made it subject to different research studies illuminating various interesting aspects. First of all, Baset and Schulzrinne [2] analyzed initial versions of Skype and revealed its different mechanisms to traverse NAT routers and firewalls. They showed that Skype is based on P2P technology and relies on the concept of Superpeers which are, e.g., used to relay calls between peers which are not able to establish a direct connection. Ehlert and Petgang [4] derived typical signatures of such relayed Skype VoIP sessions in order to support administrators in detecting Skype traffic in their network. A similar approach was applied in [5] by performing measurements on both the client and the server side of a relayed call in order to characterize and detect relayed traffic. Guha et al. [6] studied the session times and bandwidth consumption of Superpeers in the Skype overlay. Based on measurements on a specific Superpeer they derived the complementary distribution function for relayed VoIP call durations as well as for the size of files transferred over the Skype overlay.

In [7], first concerns where expressed to use Skype in a corporate environment due to security issues. The topic of security was further discussed in [8–10]. Skype encrypts its calls using AES with a block size of 128 bit and a key size of 256 bit. Authorization is done using RSA keys of up to 2048 bit. A closer look at the Skype binary also revealed that it tries to protect itself from being reverse engineered by refusing to start when tools like the Soft-Ice debugger are present.

In contrast to all previous work, we intend to study how Skype reacts to changes in the network and how this affects the satisfaction of the user with the service. Therefore we characterize the traffic generated by a Skype client in different environments and relate it to the quality as perceived by the end user. The work which comes closest to our studies is [11], in which the authors derive a User Satisfaction Index (USI) which translates typical network parameters as well as measured call durations into a performance measure for user satisfaction. The two main points in which we differ from this approach are that we regard a mobile UMTS environment and try to uncover how Skype performs in such situations and how it is able to maintain the measured user satisfaction even under the changing network conditions which are typical in mobile networks. In addition, we measure the QoE using the established and generally accepted MOS value, which relies on the comparison of audio files instead of trying to translate network parameters into user satisfaction.

Finally, there are additional aspects like video calls or the interaction between IP and regular telephony using SkypeOut or SkypeIn which are not yet studied but offer interesting opportunities for further investigations.

3. End-to-end QoS and QoE performance objectives

The end-to-end quality of the communication between two end hosts can be evaluated on different levels and from different points of view. The traditional approach captures the QoS using measurements on the network layer. The derived technical parameters precisely describe the current ability of the network to provide a service but do not necessarily reflect the quality felt by the user of the service. On that account a new paradigm emerged which intends to asses the QoE describing the satisfaction of a user with the service. In this section we give a more detailed description of the terms QoS and QoE with regard to VoIP calls, show how they are related to each other, and how we measured them in our testbed environment.

3.1. Network QoS parameters

The key factors influencing the QoS on network layer are the achieved throughput, the packet loss, the delay, and the variation of the delay which is known as jitter. During UMTS delay measurements we experienced one-way delays which were smaller than the 150 ms recommended by the ITU-T [12] for VoIP calls. In this work, we therefore concentrate on those parameters which mainly impact the QoE. To derive the throughput of a VoIP connection we measure the payload as well as the UDP and IP headers which were sent or received during an interval of Δt, respectively. Note that the headers of the data link layer and the physical layer are neglected in order to be able to compare the throughput achieved in the different wired and wireless scenarios regarded in the following sections. We also distinguish between the mean throughput m_{sent} generated by the sender and the mean goodput m_{rcvd} actually observed by the receiver during Δt. The parameters s_{sent} and s_{rcvd} describe the corresponding standard deviations. The packet loss p_{loss} can be defined in various different ways. The simplest method is to describe it as the percentage of packets lost during an interval of time Δt. We therefore start our measurements with the first actually transmitted packet, count the number of unsuccessfully transmitted packets per Δt, and divide this value by the total number of packets sent.

Finally, jitter can be described by the variation of the packet interarrival time (PIAT) at the receiver side, i.e. by the variation of the time difference between two consecutive packet arrivals. In our case, the Skype application deterministically sends its voice packets every $\Delta t_{\text{Skype}} = 30$ ms or 60 ms depending on the currently applied codec. In practical measurements Δt_{Skype} might slightly deviate from those theoretical values. In the following the time between two packets generated by the sender is denoted as packet intersent time (PIST). To capture the jitter introduced by the network, we define jitter j as the standard deviation of the PIAT at the receiver divided by the actually measured mean of the PIST:

$$j = \frac{std(\text{PIAT})}{E[\text{PIST}]}. \quad (1)$$

For the sake of readability, the term *packet interarrival time (PIT)* subsumes the PIST at the sender and the PIAT at the receiver. In more general scenarios the jitter can also be defined based on a moving average as described in [13].

3.2. Objective user-experienced quality of VoIP calls

In general the term QoE is a measure for how satisfied the end user is with the quality, the usability, and the performance of the service at hand. A common way to derive a numerical indication for the QoE of VoIP calls is to compare the quality of the original audio file to the quality of the transmitted audio file as received by the communication partner. The two most prominent algorithms are the Mean Opinion Score (MOS) [14] and the Perceptual Evaluation of Speech Quality (PESQ) [15] method described in ITU-T P.862. PESQ was specifically developed to be applicable to end-to-end voice quality testing under real network conditions, like VoIP, POTS, ISDN, GSM, etc. It takes the originally sent audio wav-file as well as the received wav-file as input and returns a value between −0.5 (worst) and 4.5 (best) which describes the satisfaction of a user with the quality of the voice call. The resulting PESQ value can be mapped into a subjective MOS value according to ITU-T Recommendation ITU-T P.862.1 [16]. The MOS value itself expresses the QoE of a voice call and can take the following values: (1) bad; (2) poor; (3) fair; (4) good; and (5) excellent.

Finally, to establish the link between QoS and QoE, the latter can be expressed as a function of the former:

$$\mathrm{QoE} = \Phi(\mathrm{QoS}) = \Phi(I_1, I_2, \ldots, I_n). \tag{2}$$

Thereby the n influence factors $I_j, 1 \leqslant j \leqslant n$ represent typical network performance measures like packet loss, delay, and jitter as mentioned in Section 3.1. An example for Φ is the computational E-model [17] that uses measured network parameters to predict the subjective quality of voice calls. The output of the model is the transmission rating factor (R-factor), which lies between zero (poor) and 100 (excellent) and can easily be mapped to a MOS value. A comprehensive overview of the E-model can be found in [18].

3.3. General measurement setup and scenarios

To measure the QoS and the QoE of Skype VoIP traffic in an UMTS environment, we set up a simple testbed as shown in Fig. 1. We installed Skype on two end hosts A and B and used different network entities and access types to establish an end-to-end connection between the two hosts. The general measurement setup is as follows: Skype user A transmits a standard audio file to Skype user B. We used an English spoken text without noise and a sample rate of 8 kHz which was encoded with 16 bit per sample and is available at [19]. The wav-file was played with the Winamp audio player on machine A, whereas the output of Winamp was used as input for Skype (instead of a microphone). Both sender A and receiver B were running Windows XP. Packet traces were captured using the latest version of Windump on each machine.

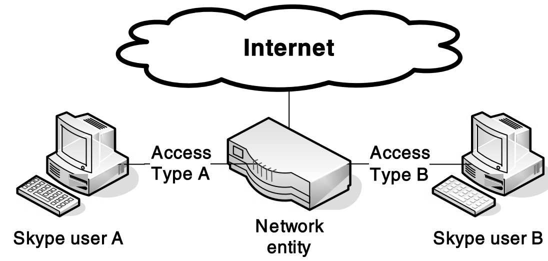


Fig. 1. General measurement setup.

Skype offers the possibility to display technical information about an ongoing call as indicated in Fig. 2. From this we learned that it selects its audio codec according to the performance of the hardware it is running on. Since we were interested in typical UMTS measurements we used CPUs with 500 MHz on host A and host B which roughly reflects the processing power of actual mobile UMTS devices and forces Skype to use the simple iLBC codec. To gain insight into the behavior of future generation mobile devices, we replace both machines in later measurements with state of the art hardware which allows Skype to use its adaptive multirate codec iSAC. Note, that the processing power of the user equipment is in no way the limiting factor in our measurements. The different hardware at the end hosts is merely used to force Skype to use its different codecs.

The main focus of our studies is on how the current network conditions influence the QoE of the end user and in how far the Skype application reacts to quality degradations. In this context, the network entity located in the middle of our testbed is used to emulate typical problems in wireless network environments. In particular, we investigate to what

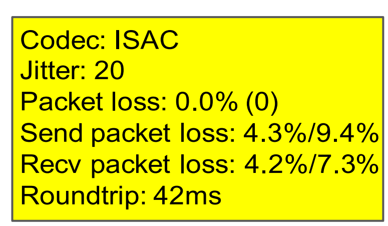


Fig. 2. Excerpt of the technical information shown by Skype.

extent the results depend on the way the network is emulated and if there are differences to measurements in a real UMTS networks. Therefore, we apply two different emulation approaches, one based on hardware and one based on software. Additionally, we perform measurements in the German UMTS network. For the hardware-based approach we used a Cisco 3660 router running IOS 12.0, the software-based approach was realized using dummynet [20] and NIST Net [21], two freely available tools which offer different abilities to emulate typical network behavior for individual end-to-end connections. Additionally, the network entity was connected to the Internet which is necessary for Skype to run on the end hosts. If not stated otherwise, we used Skype Version 1.20.37 running on Windows XP.

Our testbed environment enables us to obtain packet traces from which we can then derive end-to-end QoS models and traffic models for Skype. Comparing the recorded audio files at host B to the original audio files sent by host A we are also able to measure the QoE of Skype VoIP calls. To obtain a reference PESQ value for our measurements we encoded the original audio file (optimal PESQ of 4.5) with the iLBC codec which results in a slight degradation of the voice quality and a PESQ value of 4.02. However, the Skype application is well secured against being evaluated. It refuses, e.g., to start if a debugger is installed on the system [9]. Similarly not all versions of Skype allowed us to directly record its audio output. In such cases we forwarded the audio signal to another machine using an audio cable as shown in Fig. 3. This resulted in another slight degradation of the voice quality and a PESQ value of 3.93.

Based on the obtained QoS and QoE measurements in our testbed, we answer the following questions in the next sections:

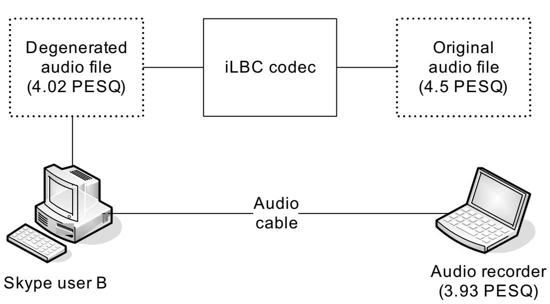


Fig. 3. Degradation of the PESQ value caused by the measurement methodology.

– Does Skype work properly with a rate-controlled DCH in UMTS?
– In how far does the emulation tool influence the behavior of Skype?
– Which impact does the UMTS network itself have on the voice quality?
– While in a call, does Skype react to network changes?

4. Emulated rate-controlled DCH in UMTS

In UMTS systems, the conditions of the wireless channel are changing over time because of radio propagation effects or fading.On rate-controlled dedicated channels (DCHs), this results in a slow adaptation of the bandwidth currently assigned to the user [22]. In order to analyze whether such DCHs suffice to carry Skype VoIP calls, we regard a simplified LAN scenario. In particular, we use our Cisco router as the traffic shaping network entity depicted in Fig. 1 and emulate the dynamically changing conditions of the DCH by restricting the bandwidth of the ongoing Skype call. Initially, we increased the bandwidth from 16 kbps to 32 kbps, 64 kbps, 128 kbps, and 384 kbps, respectively. Since during our measurements we observed that the measured PESQ values are very sensitive to small changes in the range between 16 and 64 kbps, we also measured 24 kbps, 28 kbps, 40 kbps, 48 kbps, and 56 kbps.

Depending on the currently available bandwidth and processing power, Skype uses different codecs to maintain reasonable call qualities [2]. In this measurements we assumed mobile devices with up to 500 MHz, which forced Skype (Version 1.20.37) to use iLBC [23], a simple audio codec with a fixed packet size and a fixed PIST. Since this particular version of Skype did not allow us to record its output directly, we forwarded the audio to a separate recorder according to Fig. 3. In order to evaluate and compare the perceived voice quality for the different bandwidth values, the same audio data was transmitted for each measured bandwidth.

4.1. Characterization of the iLBC codec

The throughput achieved by VoIP calls using Skype's iLBC codec is shown in Fig. 4 in dependence of the available bandwidth. It includes the payload (67 byte) as well as the UDP and IP headers (28 bytes). Each scenario was repeated between five and ten times in order to produce credible

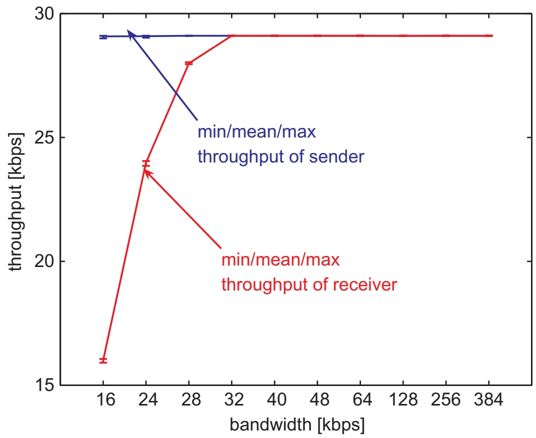


Fig. 4. Rate-controlled DCH scenario: throughput using Skype's iLBC codec.

emulation results. The figure shows the mean values of the different emulation runs as well as the corresponding minimum and maximum. Skype did neither adapt the sending rate to the available bandwidth nor to the resulting packet loss in any of the emulation runs. The sender constantly uses a bandwidth close to 26 kbps independent of the quality of the communication channel. The communication partner receives a throughput, which corresponds to the currently available bandwidth on the link, the remaining packets are lost on the bottleneck link.

In order to understand the details of the bottleneck in this scenario, we focus on a single emulation run. Fig. 5 shows the CDF of the packet interarrival time for both the sender and the receiver of the VoIP call, using a bottleneck speed of 16 kbps. The sudden jump from 0 to 1 at the CDF of the sender illustrates an almost constant time of 30 ms between two sent packets. At first glance, the CDF of the receiver has a very unexpected shape. About 90% of all packets have an interarrival time of practically 0 ms, while the time between the remaining packets is about 500 ms. This behavior is explained in the following. The buffer in the router was set to 8000 bit, while simultaneously limiting the speed of the link to 16 kbps. Skype used a total packet size of 872 bit. Thus, at most 9 packets ($872 \cdot 9 = 7848$ bit) fit into the buffer of the router. To emulate a link speed of 16 kbps, the router fills its buffer and delays the data for exactly 500 ms. This way, a bandwidth of $8000\ \text{bit}/500\ \text{ms} = 16$ kbps is achieved on a physical 100 Mbps link. This has two major implications. At first the interarrival time of the packets within a burst is 872 bit/100 Mbps, which is in the range of $10^{-6}\ \text{s} \approx 0$ ms and explains the shape of the CDF in Fig. 5. Secondly, packet loss occurs in bursts during the 500 ms, in which the buffer of the router is delayed. In Section 5 and 7, we will see that Skype adapts its bandwidth usage to packet loss as soon as it no longer occurs in bursts but randomly.

4.2. Relationship between end-to-end QoE and packet loss

To evaluate the speech quality as perceived by the end user, we have a closer look at the MOS value for the emulation runs described in the previous section, i.e. between five and ten emulation runs per scenario. Fig. 6 illustrates the achieved MOS values (cf. left y-axis) for different link speeds between 16 and 384 kbps and relates them to the

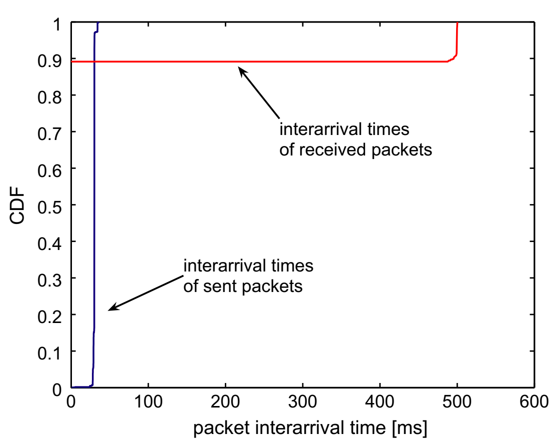


Fig. 5. Rate-controlled DCH scenario: PIT for bottleneck restriction to 16 kbps.

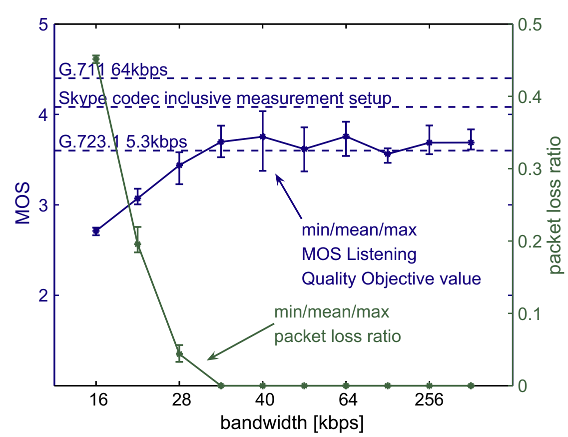


Fig. 6. Rate-controlled DCH scenario: MOS value related to packet loss.

observed packet loss (cf. right y-axis). The higher the packet loss, the lower is the corresponding MOS value. Nevertheless, the quality is sufficient to enable mobile VoIP via rate-controlled DCHs.

There is no more packet loss above a link speed of 29 kbps since up from this point the throughput of the sender (109 byte/30 ms) is smaller than the available bandwidth on the link. The corresponding MOS values oscillate around a value of 3.8. Since the MOS value is a very sensitive performance measure, the fluctuations can be explained by the stochastic influences of the network, like jitter. For a better classification, the figure also shows reference MOS values for the G.711, the G.723.1, as well as for Skype's iLBC codec, when evaluated locally. Transmitting the audio packets over the network obviously results in a slight degradation of the MOS value achieved by the iLBC codec.

5. Replication of voice data

The scope of this section is to investigate if the applied network emulator and the character of packet loss affects the behavior of the Skype application. When using a traffic shaping router to restrict the link bandwidth as in the previous scenario in Section 4, Skype did not react to the occurring packet loss and the sender's throughput did not change. This hardware-based network emulation caused a high autocorrelation of lost packets, as the limited size of the queue deterministically determines which packets get lost. This approach should be used to simulate losses due to congestion. In this section, however, we consider a lossy link with independent and random packet losses generated by a software emulator. In such scenarios, Skype shows a different traffic profile which will be characterized in the following. This particular application behavior is generalized and evaluated analytically in order to study the impact it has on the perceived QoE.

5.1. Burst losses vs. randomly lost packets

The dummynet software [20] is a popular and easy-to-use application for emulating queue and bandwidth limitations, delays, or packet losses. It works by intercepting communications of the IP layer and emulating the desired effects. It is described in detail in [24]. The dummynet software is installed on a BSD Linux machine which also acts as gateway for both machines of Skype user A and B according to Fig. 1. To emulate a lossy link between A and B, the individual packets are dropped at random with probability p_{loss}, where $p_{\text{loss}} = 0$ means no loss and $p_{\text{loss}} = 1$ makes all packets be dropped. The packet loss value can be dynamically changed during the VoIP call. For the machines of the Skype users, we use the same hardware and software configuration as in the previous Section 4.

Fig. 7 shows the throughput of the sender and the throughput at the receiver over time. In the considered scenario, we start with no loss. Due to the same configuration of the end users, the iLBC codec is used for encoding the audio data. Every 30 ms a packet of fixed size is sent. In this measurement scenario, however, the payload of a packet is 58 byte (instead of 67 byte in Section 4). After roughly 90 s, p_{loss} is set to 70%. Skype user A still sends with $m_{\text{sent}} = 22.93$ kbps including the UDP and IP header of 28 bytes, while user B only receives $m_{\text{rcvd}} = (1 - p_{\text{loss}}) \cdot m_{\text{sent}} = 6.88$ kbps on average. However, after another 25 s, Skype reacts to the detected packet loss and increases the bandwidth of the sender to $m_{\text{sent}} = 8 \cdot (115 + 28)$ byte/30 ms $=$ 38.13 kbps by changing the payload of a packet to 115 byte. As a result, the received goodput increases accordingly. We decreased the packet loss value over time to 50%, 10%, and 0%, respectively, whereas, the sender's throughput only then switched back to the original 22.93 kbps when no more packet loss was detected.

This implies that Skype sends redundant information to overcome the effects of a lossy link and to maintain a certain QoE. Considering the payload of both packets, we see that the data information is nearly doubled. The simplest approach to send

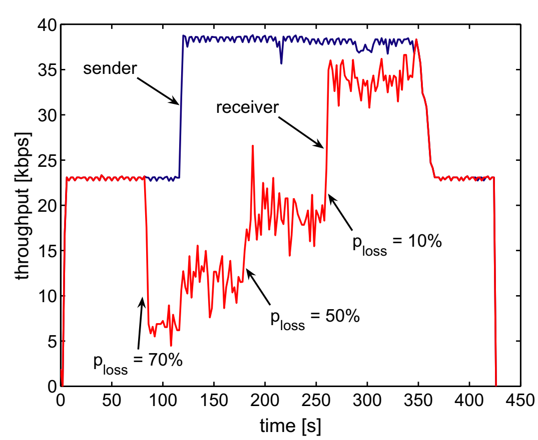


Fig. 7. Bandwidth adaptation for random losses (using iLBC).

redundant information is to replicate the entire voice information of a single audio frame and put it into two consecutive IP packets. This observation is the motivation to evaluate in general what impact the replication of an audio frame in R consecutive packets has on the end-to-end QoE (cf. Section 5.2).

The results of this experiment show that Skype reacts differently in different scenarios. As shown in Section 4.1, cf. Fig. 4, Skype does not react to burst losses and keeps sending with a constant throughput. For randomly lost packets, however, Skype adapts its bandwidth usage at runtime, cf. Fig. 7. This means Skype recognizes and distinguishes between the different reasons for packet loss, like congested or lossy links. This is referred to as edge-based intelligence. The question is on what grounds Skype decides how to react? One possibility is the distance L between two consecutive packet losses. L is referred to as inter-packet-loss distance. The number of consecutive packets without any packet loss is denoted as $K = L - 1$. In the dummynet scenario, the random variable L follows a geometric distribution shifted by one: $L \sim \mathrm{GEOM}_1(q)$ with parameter $q = \frac{\mu-1}{\mu}$ and a mean measured distance μ. When using the traffic shaping router, losses occur in a bulk. Hence, the probability that the inter-packet-loss distance is one is very high, in our scenario $P(L = 1) = 0.87$.

Another possibility is to use the autocorrelation of the observed packet loss. If we have random packet losses, the autocorrelation is close to zero for an arbitrary lag. If the losses occur in a bulk however, a high positive correlation can be detected for appropriate lags. Hence, a regular pattern of the autocorrelation for different lags can be observed, cf. Fig. 8.

Summarizing, Skype implements an edge-based intelligence to react to packet loss. The specific network characteristic generated by the applied measurement setup has a significant influence on the observed traffic profile. When detecting independent and random losses, redundant information is sent to maintain a certain QoE. This observation was the starting point for the investigation of dynamic changes during a VoIP call, which will be discussed in detail in Section 7.

5.2. Analytical evaluation of the impact of replication on QoE

Based on our experiences gained from Skype, we propose the replication of voice datagrams as a pos-

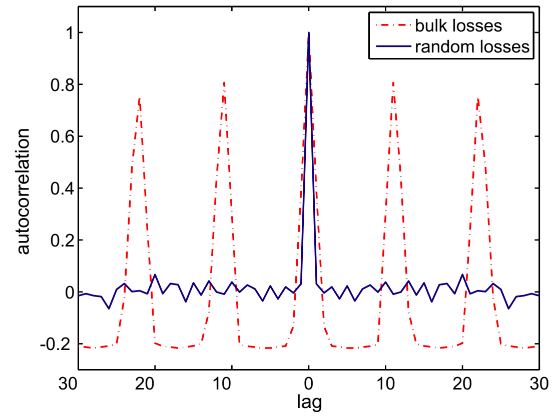


Fig. 8. Autocorrelation of packet loss for different lags.

sible solution to overcome a QoE degradation due to packet loss. This is the simplest approach to smoothen the effect of packet loss. We assume the iLBC voice codec, i.e. every $\Delta t = 30$ ms, a voice datagram of size $s_{\mathrm{voice}} = 400$ bit is sent. A *replication degree* R signifies that the voice datagram is additionally sent in the following $R - 1$ packets.

As a consequence, each packet now contains R voice datagrams with a total packet size of $s_{\mathrm{packet}} = s_{\mathrm{header}} + R \cdot s_{\mathrm{voice}}$. The variable s_{header} denotes the overhead for each packet caused by UDP and IP headers (8 byte + 20 byte) as well as by the link layer (e.g. 14 byte for Ethernet). Hence, the required bandwidth is a linear function in R:

$$C_{\mathrm{req}} = \frac{s_{\mathrm{header}} + R \cdot s_{\mathrm{voice}}}{\Delta t}. \quad (3)$$

The advantage gained by this bandwidth consumption is the reduction of the effective voice datagram loss probability $1 - p_{\mathrm{voice}}$. For a given packet loss probability p_{loss} and a replication degree R, a voice datagram only gets lost if all R consecutive packets containing this voice datagram get lost. Thus, the probability p_{voice} that a voice datagram is successfully received is

$$p_{\mathrm{voice}} = 1 - p_{\mathrm{loss}}^R. \quad (4)$$

In [25], we derived a relationship between the effective voice datagram loss probability and the obtained MOS value. This relationship is used in the following numerical example to illustrate the improvements in the achieved QoE which can be gained by the voice datagram replication. For $R = 1$ and $p_{\mathrm{loss}} = 0.2$ we obtain $p_{\mathrm{voice}} = 0.8$ and a MOS value as low as 2.29. Increasing the replication degree to $R = 2$ and $R = 3$ leads to $p_{\mathrm{voice}} = 0.96$ and

$p_{\text{voice}} = 0.992$, respectively. The corresponding MOS values are 3.63 and 4.04, respectively. This shows that the QoE could be improved from a poor quality to a good quality. A further increase of the replication degree only yields a small gain compared to the growth of the required bandwidth C_{req}.

Besides the increased bandwidth consumption, however, the replication also causes some jitter, as the voice datagrams are not received every $\Delta t = 30$ ms, but may only be successfully transmitted in one of the $R-1$ following packets. We therefore quantify the jitter by computing the probability $\tilde{y}(i)$ that a voice datagram is successfully transmitted in the ith try.

$$\tilde{y}(i) = p_{\text{loss}}^{i-1} \cdot (1 - p_{\text{loss}}). \tag{5}$$

The probability that a voice packet is received follows as

$$\begin{aligned} p_{\text{voice}} &= \sum_{i=1}^{R} \tilde{y}(i) \\ &= (1 - p_{\text{loss}}) + p_{\text{loss}}(1 - p_{\text{loss}}) + \dots \\ &\quad + p_{\text{loss}}^{R-1}(1 - p_{\text{loss}}), \end{aligned} \tag{6}$$

which agrees to Eq. (4). The number Y of trials which is required to successfully transmit a voice datagram is a conditional random variable. It follows a shifted geometric distribution and is defined for $1 \leqslant i \leqslant R$:

$$Y \sim \frac{\text{GEOM}_1(p_{\text{loss}})}{p_{\text{voice}}} \tag{7}$$

with

$$y(i) = \frac{\tilde{y}(i)}{p_{\text{voice}}} = \frac{p_{\text{loss}}^{i-1} \cdot (1 - p_{\text{loss}})}{1 - p_{\text{loss}}^{R}}. \tag{8}$$

We define the jitter j to be the standard deviation $\sqrt{\text{Var}[t_{\text{rcvd}}]}$ of the interarrival time of received packets, normalized by the average time Δt between any two sent packets,

$$j = \frac{\sqrt{\text{Var}[t_{\text{rcvd}}]}}{\Delta t}. \tag{9}$$

For the sake of simplicity, we assume a deterministic inter packet sent time and a deterministic delay $t_{s\to r}$ from the sender to the receiver. With $t_{\text{rcvd}} = Y\Delta t$ the jitter can – after some algebraic transformations – be expressed as

$$\begin{aligned} j &= \frac{\sqrt{E[(Y\Delta t)^2] - E[Y\Delta t]^2}}{\Delta t} = \sqrt{E[Y^2] - E[Y]^2} \\ &= \sqrt{\frac{p_{\text{loss}}}{(p_{\text{loss}} - 1)^2} - \frac{p_{\text{loss}}^R \cdot R^2}{(p_{\text{loss}}^R - 1)^2}}. \end{aligned} \tag{10}$$

Fig. 9 shows the jitter j for replication degrees $1 \leqslant R \leqslant 6$ in dependence of the packet loss probability p_{loss}. Eq. (10) is an exact formula, which we also validated by implementing a simulation. The solid lines correspond to the analytical calculation of the jitter, while the dashed lines show the simulation result of a (short) single run. Both curves agree and the confidence intervals of several simulation runs are too small to be visible.

The cost of the voice datagram replication – besides the increased bandwidth consumption – is an increased jitter. However, jitter also impacts the QoE and is of course one impairment factor in Eq. (2). As a result, a maximal degree $R_{\max}$ of replication exists and a further increase does not improve the QoE anymore. ITU-T G.114 recommends a latency of the end-to-end delay of 150 ms, referred to as toll quality, and a maximum tolerable latency of 400 ms. According to the end-to-end delay $t_{s\to r}$ and the inter packet sent time $\Delta t = 30$ ms, the following inequation must also hold

$$R \cdot \Delta t + t_{s\to r} < t_{\max} \tag{11}$$

for a maximum allowed latency $t_{\max}$. For example with $t_{\max} = 200$ ms and $t_{s\to r} = 10$ ms, the maximum replication degree is limited to $R_{\max} \leqslant 6$.

In general, the replication of voice datagrams is an effective way to reduce the impact of packet loss on the user perceived quality. The price to pay is a

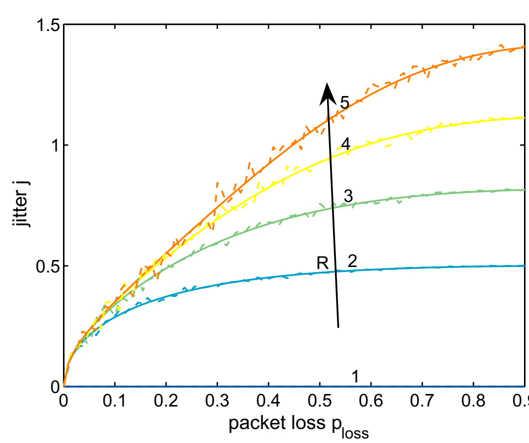


Fig. 9. Jitter depending on replication degree R and loss.

higher amount of consumed bandwidth. The benefit of the replication degree, however, is limited by the arising jitter and the higher latency of individual packets. In addition, in the analytical consideration we neglect the correlation between the consumed bandwidth and packet loss. If the packet loss is caused by congestion in the network, the additionally introduced bandwidth will worsen the network situation. An intelligent application should therefore try to estimate the cause of packet loss (e.g. using the autocorrelation as illustrated in the last section), or try to measure the effects of the increased bandwidth and to back off in case of unsuccessful counter measures.

6. Measurement in a public UMTS network

In this section, we regard UMTS scenarios where one Skype user is connected to the Internet using a public German UMTS operator. We used a Vodafone Mobile Connect UMTS PC-card as modem for the machine. During the course of the measurements, only dedicated channels (DCH) of fixed bandwidth were used. While the uplink capacity is limited to 64 kbps, the downlink direction offers a bandwidth of 384 kbps. The second Skype user is connected via DSL and has a capacity of 128 kbps in the uplink and 1024 kbps in the downlink. To account for the essential technological difference between uplink and downlink in UMTS, we have a separate look on both directions and regard two different scenarios. We investigate the *uplink scenario* in which the UMTS subscriber sends the audio data with 64 kbps to the DSL user. In the *downlink scenario* the DSL user sends its data over the 128 kbps link to the UMTS user. The measurements took place in July 2005 and each measurement scenario was repeated ten times if not stated otherwise.

During all measurements the clients used a constant-bit rate codec for the main audio connection, sending 108 byte of payload every 60 ms. Again, a derivate of the iLBC codec was used which was also indicated by the technical information shown during a call by the Skype application. However, in the UMTS scenario a different variation of this voice codec was used than in the LAN measurements in which a traffic shaping router or a software tool emulated typical UMTS network characteristics, see also Table 5. Since this codec was used starting with the first audio packet, Skype seems to choose this codec based on local information, like access type (modem or LAN) to the Internet, or due to exchanged packets before measuring the link quality. This assumption is supported by the fact, that emulating the exact link properties of the UMTS scenarios (delay, bandwidth, etc.) with dummynet did not cause Skype to use the same codec. In the UMTS scenario, there was nearly no packet loss in any of the experiments. In total, 11 out of 15417 packets were lost. However, the MOS values are lower than before because of the network jitter. In the following we concentrate on the packet interarrival times at the sender (PIST) and at the receiver (PIAT).

6.1. Uplink: UMTS subscriber sends to wireline user

In the uplink scenario the UMTS client uses a 64 kbps connection to send its data to the DSL user, which has a maximum download capacity of 1024 kbps. Fig. 10 shows the CDF of the packet interarrival time for both the sender and the receiver. The UMTS client constantly sends a voice packet every 60 ms. However, due to the jitter in the network the PIATs at the receiver side are spread around the mean. The almost symmetric shape of the CDF reflects the fact that for every delayed packet there is obviously a packet with a correspondingly smaller PIAT.

To illustrate this effect, Fig. 11 plots the PIT for each packet at the sender and the receiver. There were 878 packets in this scenario. The x-axis shows the PIT zoomed from 40 ms to 70 ms, the y-axis plots the packet of the corresponding number as sorted by their arrival time. That is, the plot shows how many packets arrived with a specific PIT. As expected, the packets of the DSL user are randomly

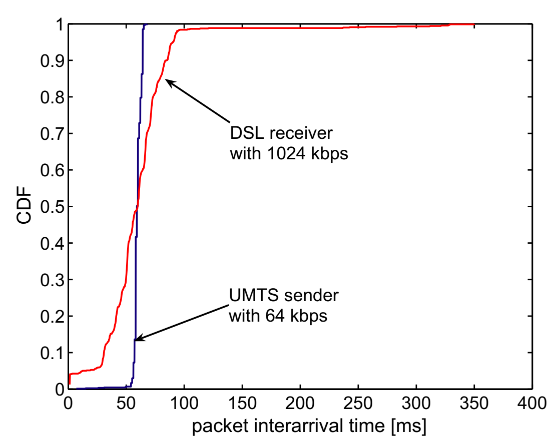


Fig. 10. Uplink scenario: CDF of packet interarrival times.

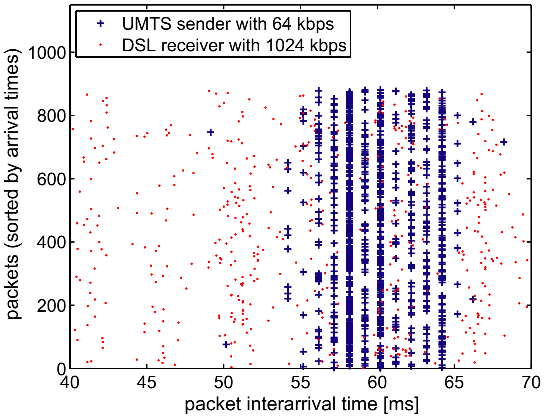


Fig. 11. Uplink scenario: PIT trace of a single run.

distributed due to the jitter in the network (red[1] dots in the figure). The UMTS packets, however, are sent at a discrete resolution of 1 ms as can be seen by the blue crosses forming vertical lines in the figure. Note that this discretization already happens at the sender and is thus locally influenced, probably by the PCMCIA UMTS card. We are therefore able to exclude buffer effects and the like in the core network for the same discretization in the downlink scenario.

Table 1 presents a more detailed view on some key performance measures for the uplink scenario using an observation window Δt of 300 ms. In particular, it shows the average throughput (m_{sent}, m_{rcvd}) and the average deviation (s_{sent}, s_{rcvd}) for ten different runs of the experiment as well as the corresponding standard deviation over the ten runs for each measure. Since there is almost no packet loss in this scenario, the throughput of the receiver corresponds to the throughput of the sender. The corresponding standard deviation (8.84 bps) of the individual runs is close to zero, as the same codec with a fixed payload size and PIST was used in each of the ten experiments. However, s_{sent} and s_{rcvd} differ by about 1200 bps in the uplink. Due to the jitter in the network the observed PIATs are almost uniformly spread around the mean PIAT, which is also reflected in the lower MOS value (3.19) as compared to the bottleneck LAN scenario (3.76) with a bandwidth restriction to 64 kbps. To highlight these effects in more detail, Table 2 shows the packets received at the DSL client during a single run of the experiment. The 3 byte packets are used for

[1] For interpretation of color in Figs. 11 and 12, the reader is referred to the web version of this article.

Table 1
Key performance measures for UMTS uplink scenario

	Throughput		Goodput		MOS
	Average m_{sent} (bps)	Deviation s_{sent} (bps)	Average m_{rcvd} (bps)	Deviation s_{rcvd} (bps)	
μ	18071.58	2300.95	18055.23	3497.57	3.19
j	8.84	568.87	21.20	858.38	0.13

Table 2
Received packets in the UMTS uplink scenario

Type	Payload (byte)	Number	Mean PIAT	Std. PIAT (ms)
A	3	3	20.02 s	10.73
B	108	847	61.97 ms	35.00
C	112	28	1.92 s	27.05
B or C	108.13	875	60.00 ms	2.55

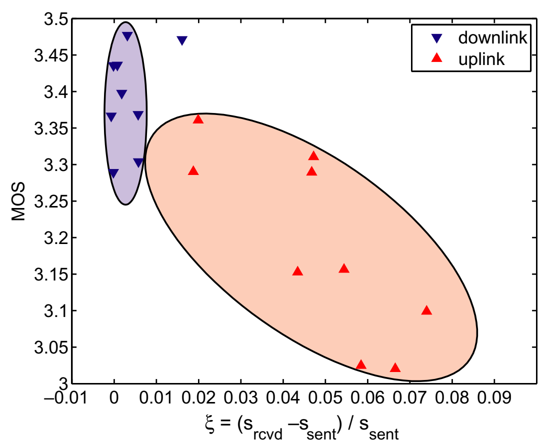


Fig. 12. MOS scatter plot as function of the jitter for UMTS.

quality feedback. However, this specific Skype codec uses two types B and C of packets (108 byte and 112 byte) in the main audio stream. Thereby every 32nd packet is of size 112 byte, which explains why the mean PIAT of the 108 byte packets is 61.97 ms instead of 60 ms. The PIAT is exactly 60 ms when we do not differ between type B and C. The high standard deviation of the PIAT of these packets confirms our previous statements. Packet type A might be used for quality feedback, packet type B is used for pure audio data, and C for audio data and signaling information. The same behavior was observed in all nine remaining experiment runs.

Fig. 12 shows the obtained MOS values for the UMTS measurements in both directions, the uplink and downlink. We used a MOS scatter plot as function of the measured throughput jitter ξ which is the normalized difference of the standard deviation of the throughput of the sender s_{sent} and the receiver s_{rcvd} for an observation window of $\Delta t = 300$ ms. The red upward-pointing triangles represent the results from the uplink scenario, the blue downward-pointing triangles the results from the downlink scenario accordingly. The higher the jitter, the lower is the MOS value. In all experiments, the uplink reveals higher jitter and hence lower quality than the downlink. In addition, the occurring jitter in the uplink shows a larger amplitude (0.01–0.08) than the downlink (0–0.006). There is only a single maverick in the downlink with a jitter of 0.0161.

6.2. Downlink: wireline user sends to UMTS subscriber

In this scenario, we regard the opposite direction, where the DSL user sends its voice data over a 128 kbps link to the UMTS user, who has a downlink capacity of 384 kbps. Thereby the interesting effects occur on the link from the base station to the UMTS mobile. Fig. 13 shows the CDF of the PIT at the DSL sender and the UMTS receiver. Like before, the packets are sent into the network exactly every 60 ms. The UMTS receiver, however, registers a different behavior of the incoming packets. The PIAT of the arriving packets is no longer uniformly spread around the mean PIAT, but mainly takes three discrete values, 40 ms, 60 ms, and 80 ms. The difference of these values, corresponds to the UMTS Transmission Time Interval (TTI) value which is typically 20 ms. As can be seen by the CDF, about 60% of all packets arrive with a PIAT of 60 ms at the UMTS receiver. Approximately every 5th packet misses the corresponding TTI, cf. Fig. 14, and subsequently arrives with a PIAT of 80 ms. Therefore the next packet, which hits the correct TTI, has a PIAT of 40 ms instead of 60 ms. This means that 20% of all packets have a PIAT of 80 ms and 40 ms, respectively.

Table 3 gives a more detailed view of the key performance measures in the downlink scenario. The mean throughput of the receiver again corresponds to the throughput of the sender. This time, s_{sent} and s_{rcvd} do not differ as much as in the uplink scenario. Thus, the network should have less influence on the user perceived quality of the audio connection. The MOS is indeed higher in the downlink scenario (3.39) than before in the uplink scenario (3.19), cf. Fig. 12. Note that the standard deviations in the last row of Table 3 are slightly higher, since the number of quality feedback packets varied in the different experiment runs.

Table 4 summarizes the four different packet sizes at the UMTS receiver in this scenario. Again for the most part 108 byte packets were used for the audio connection, while this time only every 54th packet had a payload of 112 byte. In exchange, there is a new packet type using 21 byte. This kind of packet was also used in the audio connection, replacing

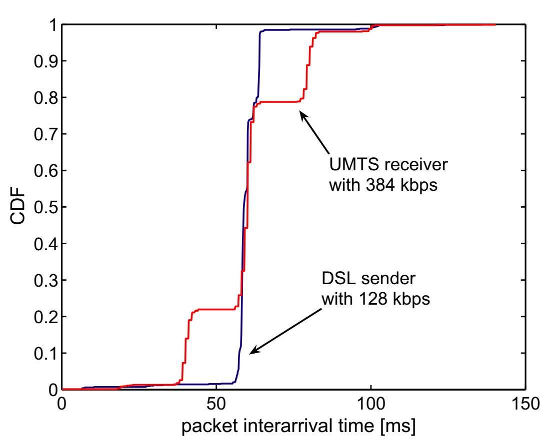


Fig. 13. Downlink scenario: CDF of packet interarrival times.

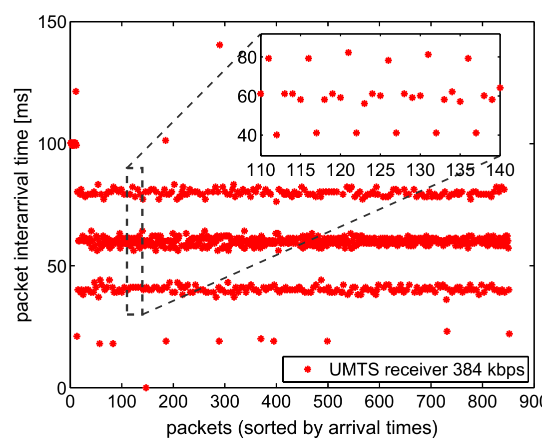


Fig. 14. Downlink scenario: PIT trace of a single run.

Table 3
Key performance measures for UMTS downlink scenario

	Throughput		Goodput		MOS
	Average m_{sent} (bps)	Deviation s_{sent} (bps)	Average m_{rcvd} (bps)	Deviation s_{rcvd} (bps)	
μ	18023.77	1848.15	18007.08	2172.39	3.39
j	48.16	282.70	51.64	284.97	0.068

Table 4
Received packets in UMTS downlink scenario

Type	Payload (byte)	Number	Mean PIAT	Std. PIAT
A	3	6	9.46 s	4.49 s
B	21	14	1.73 s	3.58 s
C	108	817	61.32 ms	16.00 ms
D	112	16	3.20 s	45.02 ms

some of the 108 byte packets. However, they were sent very irregularly as can be seen by the high standard deviation of their PIAT. The same irregularity was obtained for the 3 byte packets, which did not have a deterministic PIT of 20 s but were sent every 10 s on average with a standard deviation of 4.49 s. What exactly triggers Skype to use this specific variation of the codec is subject to further study.

7. Emulate dynamic changes in UMTS

So far, we investigated the influence of packet loss and bandwidth restrictions on a Skype VoIP call when using low power machines with a CPU power of 500 MHz. In that case, the iLBC codec was used which only requires low computational power. We have seen that Skype is an edge-based application which reacts to the network conditions in order to maintain a certain QoE.

Edge-based applications apply traffic control on application layer and thereby shift the intelligence to the edge of the network. The goal is to maintain a certain QoE, independent of the current network conditions, by performing quality control and adaptation. When the network conditions change, the application has to react on this with appropriate mechanisms. We have already seen that the voice quality of the Skype service is maintained by using appropriate voice codecs, cf. Table 5.

Currently, mobile phones with a CPU of 400 MHz are available, but in near future microprocessor units for multimedia applications will be included in the mobile devices, like the OMAP3430 [26] expected in 2007. In the following, we will use more powerful machines of 1.3 GHz in order to reveal all edge-based intelligence mechanisms and offered features of Skype. In that case, a better, but more complex codec is used, the iSAC codec, which is implemented by Global IP Sound [27]. It is a wideband, adaptive codec designed to deliver high quality sound in both high-bit-rate and low-bit-rate conditions. The adaptation of the codec is done by adjusting transmission rates to increase the listening experience for the current network situation. It requires about 6–10 MIPS.

In these measurements, we used the latest available Skype version 2.0.0.81 (February, 2006) and NistNet 2.0.12c [21], a Linux-based network emulation toolkit developed by the National Institute of Standards and Technology (NIST). To study the behavior of Skype under dynamic changes in the network, we emulate varying packet loss and different round trip times (RTT) during a VoIP call.

7.1. QoE adaptation by edge-based intelligence

First, we investigate Skype's reaction to the current packet loss of the end-to-end connection. This QoE adaptation can be illustrated by a measurement study presented in [28]. The standard audio wav-file of length 51 s is played in a loop with a pause of 9 s in between.

During the measurements we gradually increased the packet loss up to 30%, decreased it back to zero percent, and again increased it to 30% as shown on the right y-axis in Fig. 15. The left y-axis shows that the PIST of the sender remains unaffected at 30 ms (with a measured standard deviation of 6.65 ms) during the entire process. The PIAT of the receiver, however, increases and decreases according to the

Table 5
Overview on the variety of voice codecs used by Skype

Type	Payload (byte)	Interval (ms)	Bit rate (kbps)	Details
iLBC-20	38	20	15.2	
iLBC-30	50	30	13.3	
iSAC	Adaptive frames		10–32	Requires 6–10 MIPS
Skype-I (traffic shaping router)	67	30	17.9	iLBC*, Section 4
Skype-IIa (dummynet)	58	30	15.5	iLBC*, Section 5
Skype-IIb (dummynet)	115	30	30.7	iLBC*, packet loss detected
Skype-III (UMTS)	108	60	14.4	iLBC*, Section 6
Skype-IV (dynamic changes)	89–286	18–36	17.3–111.2	iSAC*, Section 7

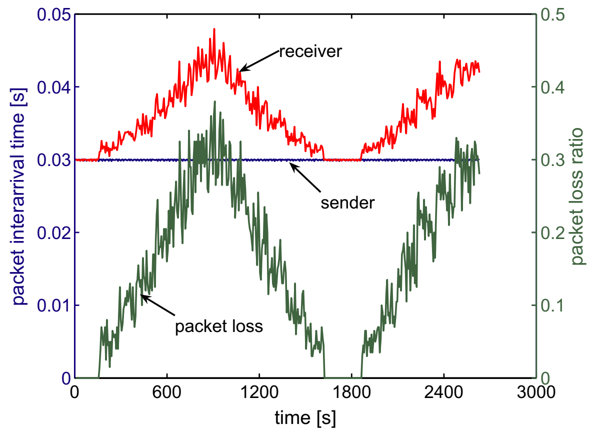


Fig. 15. Dynamic change of packet loss: measured PITs.

current packet loss rate. Fig. 16 shows how the Skype software reacts to this kind of packet loss. The measured packet loss ratio on the right y-axis denotes how many packet got lost, whereby we used the average for a window size of 6 s. On the left y-axis, the average size of the voice packets on application layer is plotted in bit. Again, we used a window size of 6 s corresponding to 200 voice packets. Initially, the Skype call is established between user A and B without any packet loss on network layer. The size of a packet varies between 90 bit and 190 bit, resulting in an average of 150 bit. The oscillations of the packet size are due to our measurement setup, as during the 9 s pause interval, Skype still sends small packets of size 50 bit.

After 5 min, we start to increase the packet loss probability by about 5% every 2 min, until the packet loss probability reaches 30%. The time interval of two minutes was chosen to ensure that Skype has enough time to react to changes. We noticed that Skype needs about 1 min to adapt its voice

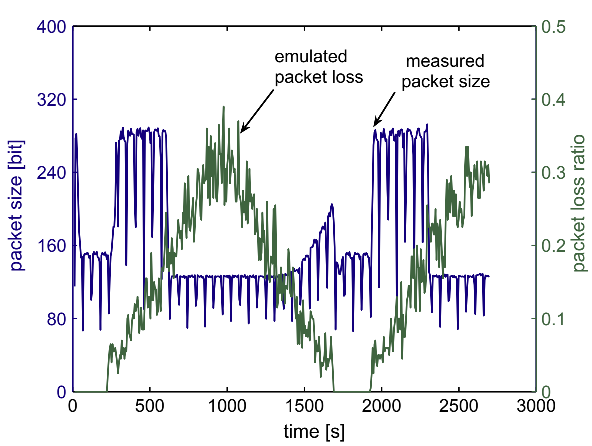


Fig. 16. Dynamic change of packet loss: measured packet size.

codec to the current network conditions. As we can see in Fig. 16, Skype reacts to the experienced QoE degradation in terms of packet loss by increasing the packet size. The new size ranges between 240 bit and 320 bit with an average of 280 bit. In contrast to before, the packet size is nearly doubled. This implies that Skype now sends redundant information within every voice packets in order to maintain the QoE. However, as soon as a certain threshold is exceeded (in this case about 20% packet loss), the packet size is decreased again to a lower average value of 125 bit as compared to the original average packet size. This indicates a change in the used voice codec. As soon as the packet loss probability falls below a certain threshold, the sender rate is again adapted by increasing the packet size.

This measurement clearly shows that Skype in fact tries to keep the QoE above an acceptable threshold. This is done by adapting the amount of consumed bandwidth. If the receiver's application detects packet loss, it instructs the sender to increase the bandwidth. For a VoIP call, this is easily possible, since the connection is full duplex and the connection from user B to user A is used to send the feedback information.

7.2. Application-driven re-routing

Finally, the impact of the round trip time on the quality of a Skype call is evaluated. Therefore, we repeatedly played the audio file five times while a constant delay from machine A to machine B is set. Note, that we only disturb the direct connection between A and B. The one-way delay d from A to B and vice versa is varied from 0 s to 4 s.

Fig. 17 shows the MOS of the audio call in dependence of the one-way delay. We plotted the minimum, the average, and the maximum MOS out of the five repetitions for each delay. There is only a small influence on the voice quality for delays smaller than or equal to 250 ms, i.e. $d \in \{0\ \text{ms}; 5\ \text{ms}; 45\ \text{ms}; 50\ \text{ms}; 100\ \text{ms}; 250\ \text{ms}\}$. This is consistent with ITU-T G.114 which recommends a latency of the end-to-end delay of 150 ms, referred to as toll quality, and a maximum tolerable latency of 400 ms [12].

After that a strong decay of the MOS from roughly 3.9 to 2.4 is observable for delays $d \in \{500\ \text{ms}; 1.0\ \text{s}; 1.5\ \text{s}; 2.0\ \text{s}\}$. Thus, the voice quality drops from good to poor. It is expected that an increase of the delay further worsens the quality. However, looking at Fig. 17, the average MOS value

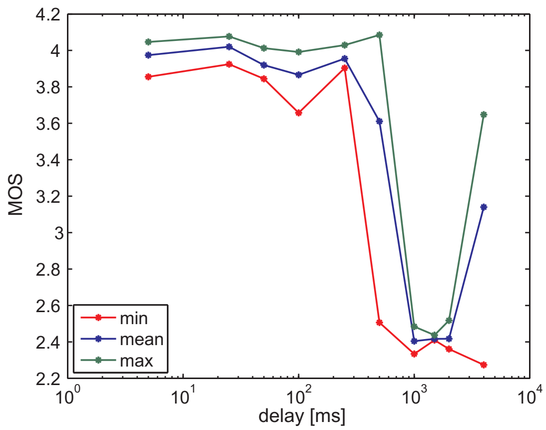


Fig. 17. Dynamic change of delay: measured MOS values.

increases again for a large delay of 4 s. The reason is that Skype relays the connection over a third-party machine, if the current connection becomes bad. Thus, Skype implements *re-routing on application layer* and forms its own logical overlay. In our measurements, Skype used a different machine C in the Internet as a relay node. After 15 s, the traffic was redirected from A to C to B, instead of the direct, but disturbed connection, from A to B.

This behavior nicely demonstrates the way edge-based applications are intended to work. The current end-to-end QoS and QoE is measured and evaluated. Performance measures may be the processing power of the involved machines or the QoS of the connection, like packet loss or delay. The application reacts accordingly, e.g. by changing the voice codec, by adjusting the sender bandwidth, or by re-routing the call on application layer. Table 5 shows the variety of voice codecs used by Skype. The payload and the packet interarrival time is related to the suggested main audio datagrams. The two basic codecs which were used during the course of the measurements are the iLBC-30 and the ISAC codec. This was also displayed by Skype's technical information field. In the different scenarios, however, different derivates of these codecs were used. Especially, the emulation of changes during a call showed the potential of Skype and the different possibilities of how to react appropriately to different network situations in order to maximize the current QoE of an user.

8. Conclusion

The Skype VoIP telephony service has become a strong competitor to existing telephone networks. In this work, the quality of experience (QoE) of Skype calls over UMTS was measured and analyzed in terms of the MOS value. Additionally, the classic quality of service (QoS) parameters like throughput or jitter are measured to derive a traffic profile for the Skype application. Different scenarios revealed that Skype is a multi-network service with edge-based intelligence, i.e. it forms a logical overlay and controls the VoIP traffic on application layer.

First, we emulated the rate control mechanisms in UMTS by restricting the link bandwidth with a traffic shaping router. In this case, Skype does not react to packet loss as caused by congestion in the network, but it constantly sends audio data. The occurring packet loss degrades the QoE, while Skype still works properly with rate-controlled DCHs. When using a software tool for emulating a lossy link, Skype generates a different traffic profile. It sends redundant information in succeeding packets, as soon as independent and random losses are detected. Hence, the edge-based intelligence tries to overcome packet loss by adapting the bandwidth in order to maximize the current QoE. The general benefit of the replication of voice datagrams was analytically investigated. The cost of the replication – besides the increased bandwidth consumption – is an increased jitter which also impacts the QoE. As a result, a maximal degree of replication can be derived up to which an increase of the QoE can be achieved. The measurements in a public UMTS network showed that the capacity offered by UMTS is sufficient to make mobile VoIP calls possible. However, due to network jitter and the use of a different codec by Skype, the MOS values are worse than those in the emulation of the bottleneck in a LAN environment. The used UMTS card sends and receives packets at discrete time instants in multiples of 1 ms. The packet interarrival times on the downlink are multiples of 20 ms, which corresponds to a common transport time interval (TTI) in UMTS.

Finally, we investigated dynamic changes in the UMTS network. One possibility to maintain the voice quality of the Skype service is the selection of appropriate voice codecs. The power of the processing unit determines whether a constant-bit rate iLBC derivate or the more complex, adaptive iSAC codec is used. Another possibility is the adaptation of the bandwidth and the replication of information to overcome packet loss, even during a call. However, if the direct end-to-end connection between two users is too poor, Skype initiates re-routing on application layer by relaying the traffic over a third-party machine. This variety of mechanisms to maximize the QoE reveals the edge-based intelligence of the Skype application. Traffic engineering in future Internet is expected to follow this new paradigm.

From an operator's point of view, it will be an increasing challenge to cope with such new edge-based applications, which are already highly popular among the users for a variety of reasons. They offer good quality, are easy-to-use, and provide additional functionality like chatting and file transfer which was not available in traditional telephony. Most importantly, the flat-rate cost models for ubiquitous Internet access additionally make VoIP very affordable. Moreover, operators will not be able to stop user-driven applications at the edge of the network, since the corresponding traffic cannot reliably be distinguished from regular IP traffic. However, as the traffic is transported via the Internet, there are no QoS guarantees like in regular circuit switched UMTS calls. Thus, if a network operator does not want to be reduced to a bit pipe, he needs to offer strict QoS and QoE guarantees and value-added services, like location-based services in mobile environments.

Acknowledgement

The authors would like to thank Melanie Brotzeller and Michael Duelli for performing the measurements.

References

[1] T. Hoßfeld, A. Binzenhöfer, M. Fiedler, K. Tutschku, Measurement and analysis of skype VoIP traffic in 3G UMTS systems, in: Proceedings of IPS-MoMe 2006, Salzburg, Austria, 2006.

[2] S.A. Baset, H. Schulzrinne, An analysis of the skype peer-to-peer internet telephony protocol, in: Proceedings of INFOCOM 2006, Barcelona, Spain, 2006.

[3] ITU-T Recommendation P.800.1, Mean Opinion Score (MOS) Terminology, 2003.

[4] S. Ehlert, S. Petgang, Analysis and signature of skype VoIP session traffic, Tech. Rep., NGNI-SKYPE-06b, Fraunhofer FOKUS, Berlin, Germany, 2006.

[5] K. Suh, D.R. Figueiredo, J. Kurose, D. Towsley, Characterizing and detecting relayed traffic: a case study using Skype, in: Proceedings of INFOCOM 2006, Barcelona, Spain, 2006.

[6] S. Guha, N. Daswani, R. Jain, An experimental study of the skype peer-to-peer VoIP system, in: Proceedings of IPTPS 2006, Santa Barbara, CA, USA, 2006.

[7] D. Bergström, An Analysis of Skype Voip Application for Use in a Corporate Environment, 2004 <http://www.geocities.com/bergstromdennis/>.

[8] S. Garfinkel, Voip and Skype Security, 2005 <http://www.skypetips.internetvisitation.org/files/VoIP%20and%20Skype.pdf>.

[9] P. Biondi, F. Desclaux, Silver Needle in the Skype, Presentation at the Black Hat Europe 2006, 2006 <http://www.blackhat.com/>.

[10] T. Berson, Skype security evaluation, Tech. Rep., ALR-2005-031, Anagram Laboratories, 2005.

[11] K.-T. Chen, C.-Y. Huang, P. Huang, C.-L. Lei, Quantifying skype user satisfaction, in: Proceedings of SIGCOMM'06, Pisa, Italy, 2006.

[12] ITU-T Recommendation G.114, Oneway Transmission Time, 1996.

[13] Audio–Video Transport Working Group, H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, RFC 1889: RTP: A Transport Protocol for Real-Time Applications, 1996.

[14] ITU-T Recommendation P.800, Methods for Subjective Determination of Transmission Quality, 1996.

[15] ITU-T Recommendation P.862, Perceptual Evaluation of Speech Quality (PESQ), An Objective Method for End-to-end Speech Quality Assessment of Narrowband Telephone Networks and Speech Codecs, 2001.

[16] ITU-T Recommendation P.862.1, Mapping Function for Transforming P.862 Raw Result Scores to MOS-LQO, 2003.

[17] ITU-T Recommendation G.107, The E-model, A Computational Model for Use in Transmission Planning, 1998.

[18] L. Carvalho, E. Mota, R. Aguiar, A.F. Lima, J.N. de Souza, A. Barreto, An E-Model Implementation for Speech Quality Evaluation in VoIP Systems, in: Proceedings of ISCC'05, La Manga del Mar Menor, Cartagena, Spain, 2005.

[19] Signalogic, Speech Codec Wav Samples, 2005 <http://www.signalogic.com/index.pl?page=codec_samples>.

[20] L. Rizzo, Dummynet Software, <http://info.iet.unipi.it/luigi/ip_dummynet/>.

[21] National Institute of Standards and Technology, NIST Net – A Linux-based Network Emulation Tool, 2006 <http://www-x.antd.nist.gov/nistnet/>.

[22] A. Mäder, B. Wagner, T. Hoßfeld, D. Staehle, H. Barth, Measurements in a Laboratory UMTS Network with time-varying Loads and different Admission Control Strategies, in: Proceedings of IPS-MoME 2006, Salzburg, Austria, 2006.

[23] S. Andersen, A. Duric, H. Astrom, R. Hagen, W. Kleijn, J. Linden, RFC 3951: Internet Low Bit Rate Codec (iLBC), 2004.

[24] L. Rizzo, Dummynet: a simple approach to the evaluation of network protocols, ACM Computer Communication Review 27 (1) (1997) 31–41.

[25] T. Hoßfeld, P. Tran-Gia, M. Fiedler, Quantification of quality of experience for edge-based applications, in: 20th International Teletraffic Congress (ITC20), Ottawa, Canada, June 2007.

[26] Texas Instruments, OMAP3430, 2006 <http://focus.ti.com/>.

[27] G.I. Sound, GIPS iSAC, 2006 <http://www.globalipsound.com/datasheets/iSAC.pdf>.

[28] T. Hoßfeld et al., VoIP Applications across Mobile Networks, in: 17th ITC Specialist Seminar, Melbourne, Australia, 2006.

Tobias Hoßfeld is a research assistant at the department of Distributed Systems, University of Wuerzburg, since 2003. His focus is on the investigation of mobile networks, in particular 3G and 4G, and on overlay techniques. He is currently involved in several industrial projects (Siemens AG, T-Mobile, France Telecom). His main research interests cover the application, the modeling, and the performance evaluation of overlay techniques in mobile networks.

Andreas Binzenhöfer studied in Wuerzburg, Germany, and at the UCLA in California, USA. Since 2002 he is a research assistant at the department of Distributed Systems at the University of Wuerzburg. He takes an active part in European Cost and NoE Projects, as well as in industrial cooperation with Siemens, BMW and Datev, where he is acting as an advisor in the Trend-Scout Project. His current research area is performance analysis of structured P2P systems and their application to distributed network management.