University of Würzburg
Institute of Computer Science
Research Report Series

The Performance of Multiplexing Voice and Circuit Switched Data in UMTS over IP-Networks

Michael Menth

Report No. 263 July 2000

Department of Distributed Systems
Institute of Computer Science
University of Würzburg
Am Hubland, D-97074 Würzburg, Germany
menth@informatik.uni-wuerzburg.de

The Performance of Multiplexing Voice and Circuit Switched Data in UMTS over IP-Networks

Michael Menth
Department of Distributed Systems
Institute of Computer Science
University of Würzburg
Am Hubland, D-97074 Würzburg, Germany
menth@informatik.uni-wuerzburg.de

Abstract

For the transmission of voice traffic in IP-networks the commonly used protocol suite for realtime data transfer in IP-networks leads to a very large protocol overhead for voice data and a poor utilization of the bandwidth for user data. Multiplexing several flows on a route into one IP packet overcomes this problem. In the future UMTS, realtime traffic like voice and circuit switched data are transported over the same network in the wireline part of the UMTS terrestrial access network. The following investigation shows the performance tradeoffs of multiplexing in two different types of quality of service enhanced IP-networks. The analytical results take into account both voice and circuit switched data. Various multiplexing and tunneling schemes are compared.

Keywords: IP, RTP, Multiplexing, UMTS, CAC, QoS, VoIP, AAL-2

1 Introduction

The success of the Internet Protocol (IP) has started the discussion in the standardization body of the 3rd generation of mobile communication systems (3GPP) to introduce IP as the transport technology in the wireline part of future mobile cellular communication systems [1]. Characteristics of realtime data traffic like compressed voice or 64 kbps circuit switched (CS) data are the strict quality of service (QoS) requirements they have, i.e., upper bounds on packet loss and delay. At the moment, both are problematic with IP technology.

A User Datagram Protocol (UDP) header is mandatory to carry information over IP networks and for voice data an additional Realtime Transmission Protocol (RTP) header is commonly used. Compressed voice samples come in small-sized packets and so tunneling, i.e., carrying a single voice sample in one IP packet yields a low bandwidth exploitation due to header overhead. This can be overcome by multiplexing several voice samples into the payload of a single RTP/UDP/IP packet. A timer is used to limit the multiplexing delay. Therefore, the RTP multiplexing scheme [2] was recently discussed by the Internet Engineering Task Force (IETF). A problem of similar nature occurs in ATM networks and is solved by using the ATM Adaptation Layer 2 (AAL-2) for multiplexing. This has been thoroughly investigated in [3, 4, 5, 6, 7].

So far, the Internet is without realtime capabilities, however, IP is currently being enhanced with realtime enabling techniques like Differentiated Services (DiffServ) [8] or Integrated Services (IntServ) [9, 10]. DiffServ is discussed with only peak rate information e.g. to realize a virtual leased line. If the traffic contract is not met for an IP packet, it might be dropped by the network, and therefore, a spacer should delay the packets to achieve peak rate conformance. IntServ requires a leaky bucket description for the traffic contract and to avoid losses by the policing unit, the bucket size must be properly set.

In a transmission system with RTP multiplexing and subsequent spacing or policing, parameters like the multiplexing timer value, the spacer buffer size or leaky bucket size have to be set properly in order to maximize the number of supportable customers for which the QoS characteristics are met. Then, a connection admission control (CAC) must shelter the users in the system from overload.

In this paper RTP multiplexing in QoS-IP-networks is investigated. Case studies are made to set the system parameters in order to maximize the performance. In Section 2 we develop a model for the transmission of compressed voice and CS data traffic. Performance measures from this model are derived in [11]. In Section 3, numerical results show the performance of tunneling and multiplexing system under various circumstances. Systems with peak rate shaping as well as with leaky bucket policing are considered. Finally, Section 5 concludes the paper.

2 Models for Wireless Data Transmission

Traffic originating from a cellular mobile communication system must be carried over a wired network to its destination. Voice data are small, therefore, tunneling causes high header overhead in IP-networks. Multiplexing several connections makes the communication more efficient. Voice and CS data are realtime applications and require realtime guarantees from the wired network part. Either peak rate spacing or a leaky bucket traffic contract can be used. Using the first one, bursty traffic must be delayed, using the second one, a leaky bucket parameter has to be declared in addition to the needed bandwidth.

2.1 Source Model

In today's cellular mobile communication systems as well as in the future Universal Mobile Telecommunications System (UMTS) (Figure 1), voice and CS data traffic originating at a mobile handset is transmitted over the radio interface to a base station (NodeB). It is transported on a wired network over the drift radio network controller (D-RNC), the serving radio network controller (S-RNC), and the UMTS mobile switching center (UMSC) into the core network and from there either into the UMTS terrestrial access network (UTRAN) or into the public switched telephone network of its peer. The wired network might consist of a virtual private network (VPN) or leased lines.

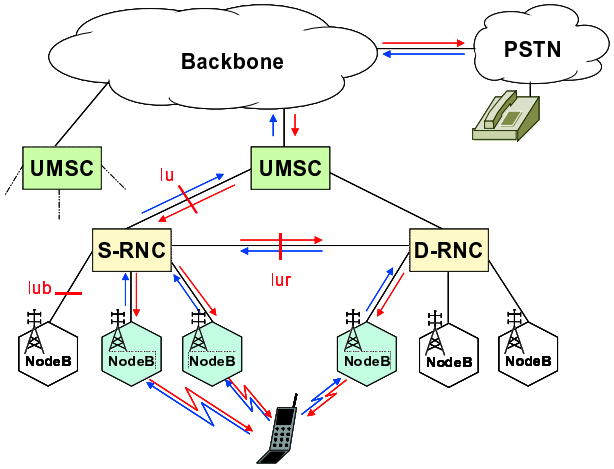

Figure 1: A soft-handover in UMTS

2.1.1 Source Model for Voice

In UMTS an audio handset transmits voice samples periodically every 20 msec. Thus, the voice arrival process is fully characterized by a single time frame of $F_v = 20$ msec. The probability that a sender is associated with an arbitrary instant within that interval is uniformly distributed. This entails an exponential distribution of the interarrival time A of consecutive voice samples if the periodic time structure is not taken into account. However, according to [12], the maximum allowed delay budget for multiplexing and sending is 1 msec. That value is still in discussion. It is only $\frac{1}{20}$ of the frame period, so that the periodical structure of the arrival process is not supposed to have great influence on the performance, especially in the presence of many users. The discrete nature of digital communication systems proposes the geometric distribution – the discrete time counterpart of the exponential distribution – to model the interarrival times of consecutively arriving voice samples.

In the considered wireless network an adaptive multi-rate (AMR) vocoder is used. During an off-phase of a conversation the information can be better compressed than during an on-phase resulting in voice samples of different size. Therefore, a sample trace of a single vocoder is clearly positively correlated.

With the argument from above – the delay budget is small compared to the periodic structure of the arrival process – for superposition of several users the sizes of consecutively arriving voice samples can be assumed to be sufficiently uncorrelated. We use a typical 3-modal distribution of the output sample size B

($\overline{B} = 18.9672$) from an AMR vocoder applied to voice traces. We model the voice sample size B by an independent and identically distributed (iid) random variable according to the given histogram.

2.1.2 Source Model for 64 kbps Circuit Switched Data

A typical application of the CS data service is a videophone conferencing, i.e. peers communicate using realtime voice and video. A data stream of 64 kbps CS data means one byte is sent every 125 μsec, however, data will be carried in packets that are assembled during a frame of length F_{CS}. With the same arguments from above, the interarrival time for n CS data users can be modeled by a geometric distribution with a mean of $\frac{n}{F_{CS}}$ and a constant packet size of $\frac{F_{CS}}{125\mu\text{sec}}$ bytes.

2.2 Data Transfer Protocol Alternatives in IP-Networks

Realtime data packets must be carried between the different network entities in UMTS. In principle, any kind of protocol could be used, but the success of the Internet Protocol in the past offers itself as a transmission technology.

2.2.1 Tunneling Realtime Data

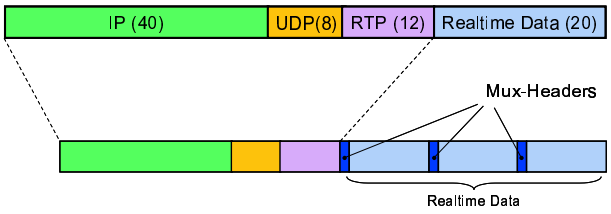

Figure 2: The protocol suite for tunneling and multiplexing realtime data.

Carrying a single realtime packet in an IP packet is called tunneling. When a realtime data packet is tunneled in the Internet, usually, the RTP/UDP/IP protocol suite is used. In the new version of the IP protocol, the header has 40 bytes and carries information about the source and destination machine. The new version (IPv6) [13] of the IP protocol will be necessary in the future when all communication devices need to be equipped with an IP address. The UDP header is 8 bytes in size and qualifies the destination port [14]. The RTP header has 12 bytes and carries informations like a timestamp or a sequence number [15]. As mentioned before, the average voice packet size is not even 19 bytes. More than 300% protocol overhead is carried by the network resulting in a low bandwidth exploitation by user data.

2.2.2 Multiplexing Realtime Data

Multiplexing is a means to reduce the overhead caused by the RTP/UDP/IP protocol suite (Figure 2). An Internet draft [2] proposes to carry several realtime packets in a single RTP-packet. Only a 2 byte multiplexing header is used to carry a 1 byte connection identifier (CID) and the realtime packet length to delimit the multiplexed data packets. Multiplexing means mapping different realtime connections to the same destination IP address and port differentiating them by the CID (Figure 3). The multiplexing process must be limited by a timer to avoid unacceptable delays for realtime data. Therefore, the timestamp of the RTP-packet approximates the ones of the carried realtime packets. Demultiplexing reconstructs the different realtime data streams. When demultiplexing and multiplexing is applied at once, e.g. to route the traffic in the UMSC or in the core, we can talk about RTP switching. We denote the multiplexing protocol described above by Mux-2-12 since it has 2 bytes multiplexing header and 12 bytes RTP header. However, if the original RTP information is needed, there is an Mux-13-12 option. One byte for the CID value might not be enough, so that we will also consider a Mux-3-12 version that allows more realtime connections to be multiplexed. If no RTP information is needed, Mux-3-0 can be used.

Another means to get rid of the large protocol overhead is header compression [16, 17] like the one that has to be applied e.g. on the air interface. However, this approach works only on a link-by-link basis since

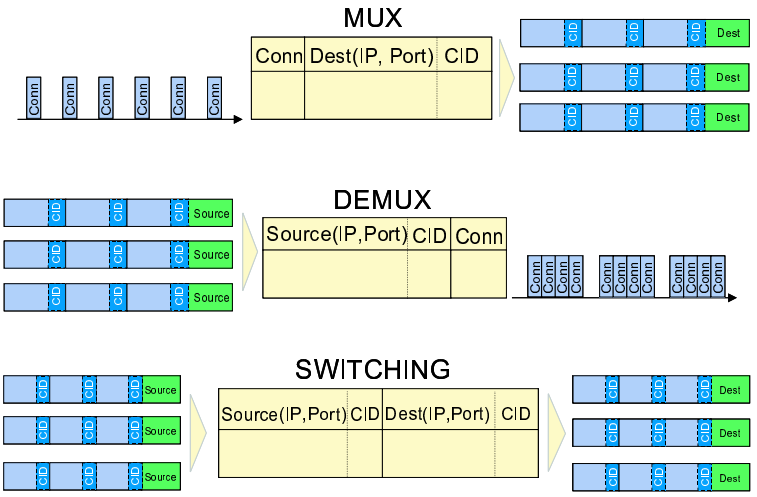

Figure 3: RTP switching performs demultiplexing and multiplexing simultaneously.

the output of a header compression is not an IP-packet any more. A multiplexed packet in contrast is an ordinary IP-packet and can be transparently routed through the Internet (Figure 4). This is an important feature, when the traffic has to be carried over the network of a different ISP.

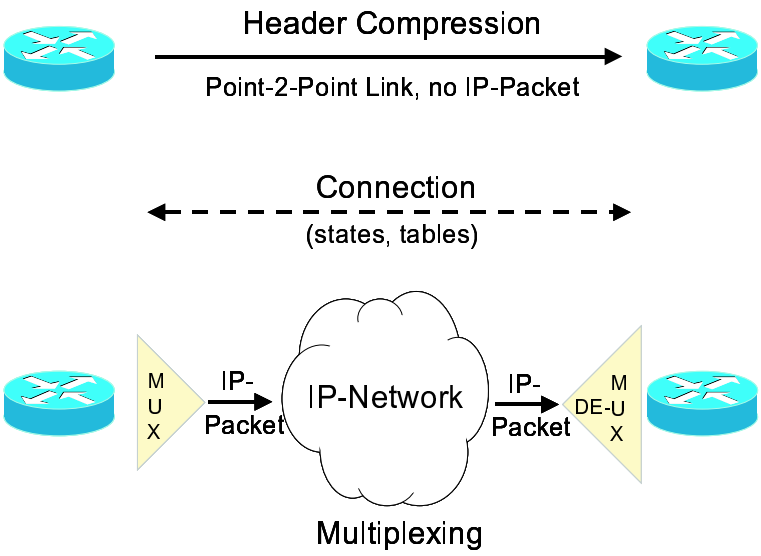

Figure 4: The difference between header compression and multiplexing.

2.3 Realtime Transfer Traffic Contracts

When the IP packet is filled, it is sent through a realtime network. Realtime transportation requires the network to dedicate enough bandwidth to the flow. To realize that efficiently, the realtime flow has to declare his traffic parameters. In return, the access must be controlled to shelter the QoS from an – intentionally or not – misbehaved source, i.e., the policing unit of the network discards packets violating the traffic contract.

In ATM, the Constant Bit Rate class [18] is commonly used and a peak rate must not be exceeded. For Integrated Services' [9] Guaranteed Quality of Service class [10] the data stream must be leaky bucket

conform which is a variable bitrate contract. Whether Differentiated Services [8] will support hard realtime constraints is not clear yet. Since peak rate spacing and leaky bucket characterization are two basic options, we will investigate both of them.

2.3.1 Peak Rate Spacing

When a traffic contract requires only a peak rate, it must never be exceeded. If a packet is found to be not conform to the traffic contract at the network boundaries of a different service provider, the policer either discards it immediately at the ingress or the packet will be marked and dropped later inside the network if congestion occurs. Therefore, packets must be spaced, i.e., deferred until the traffic contract is met. Spacing introduces additional delay which might have a considerable impact on the overall performance of the system.

The spacer that we consider works as follows. It has a byte counter S that shows the virtual occupancy of its queue. It is decreased linearly by the link rate C over time but does not fall short of zero. When an IP packet of size B arrives, it is accepted if the counter S plus the new packet's size do not exceed the queue limit S_{max}. In this case the packet will be sent after $\frac{S}{C}$ time and the counter is increased by B bytes. Otherwise, the IP packet is discarded.

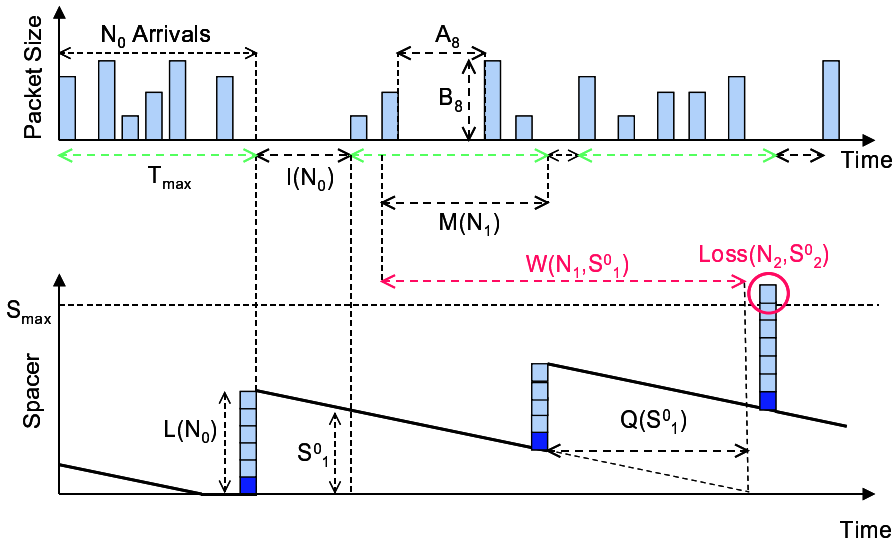

Figure 5: Multiplexing with subsequent spacing

Figure 5 summarizes the model for multiplexing realtime data with subsequent spacing. Data packets arrive with geometrically distributed interarrival times A. Their size B follows a given histogram. Then they are multiplexed using the RTP/UDP/IP protocol according to the value for the multiplexing timer TCU. The resulting IP packets are treated according to the specification of the above described spacer. The waiting time $W = M + Q$ is the sum of the multiplexing time M and the spacing time Q which is a kind of queuing time.

2.3.2 Leaky Bucket Traffic Contract

Another possibility is to describe the IP stream by the leaky bucket parameters transmission rate C and bucket size S_{max}.

A leaky bucket traffic contract allows the source to send data streams that sometimes exceed the booked bandwidth. The variation is controlled by a leaky bucket policer at the ingress of the IP-network to protect it from illegal overload. A leaky bucket policer can be seen as a virtual spacer that doesn't delay the packets but discards them if the spacer counter exceeds the spacer size which is now called the bucket size. Using a leaky bucket traffic contract, the IP packets can be sent immediately, hence, there is only multiplexing delay. To avoid losses, the leaky bucket size must be set large enough, but to save transmission costs it has to be minimized.

2.4 QoS Criteria and Performance Measures

We define the loss probability of at most 10^{-6} a QoS criterion for a realtime packet. According to [12] the delay must not be larger than $D = 1$ msec. Since this is very restrictive, we define a second QoS criterion: the probability of a voice sample waiting longer than the delay budget must be at most 10^{-4} (Figure 6). These are the constraints for all data computed in the following section.

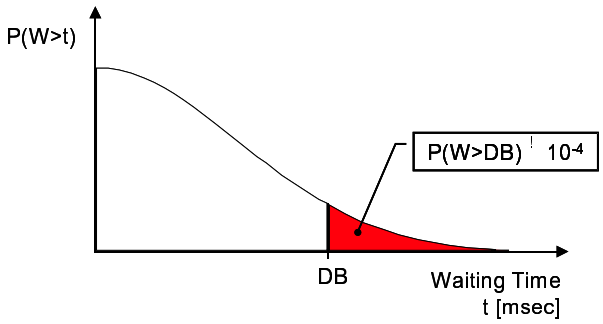

Figure 6: The quantile of the waiting time

The net user data bandwidth is denoted by $C^* = n_v \cdot \frac{\overline{B_v}}{F_v}$ where n_v is the number of voice users, $\overline{B_v}$ the mean voice sample size, and F_v the frame length for voice in UMTS. This can be extended to CS data and a mixture of voice and CS data.

The offered load $\rho^* = \frac{C^*}{C}$ is the fraction of the net user data bandwidth C^* divided by the link bandwidth C. The overall link utilization ρ is then computed by $\rho^* \cdot (1 + oh)$ where oh is the resulting voice sample overhead. The number of supportable calls can be computed by $n_v = \frac{\rho^* \cdot C \cdot F_v}{\overline{B_v}}$, again, this formula can be generalized for any mix of voice and CS data.

3 Results

The numerical results presented in this section are gained from an analysis derived in [11]. The system is described in discrete units. The spacer counter S and the packet size B are measured in bytes while time is measured in discretized time units (TU). For the interarrival time of consecutively arriving realtime data packets the geometric distribution seems to be appropriate for the voice transmission model on the access link. It can be scaled by $\overline{A} = \frac{\overline{B}}{C \cdot \rho}$. The coefficient of variation is $c_{var} = \sqrt{\frac{\overline{A}-1}{\overline{A}}}$. If $\overline{A}$ is close to 1, i.e., there are many arrivals in the system, it is heavily loaded or the packets are very small, the coefficient of variation is close to zero, otherwise it is bounded by 1.

In the following, performance studies are performed both for peak rate spacing and leaky bucket traffic contracts. The transmission of voice and CS data is investigated and tunneling is compared with its multiplexing alternatives.

3.1 Peak Rate Spacing

As mentioned before, a traffic contract that comprises only the peak rate requires the source to space the data stream to prevent uncontrolled losses at the ingress of the carrying QoS-IP-network due to policing. The multiplexing time and the spacing time contribute both to the realtime packet waiting time. If the delay budget QoS criterion can be met, the loss criterion can also be fulfilled just by having a large enough spacer buffer. Hence, the limiting factor for the critical load is the realtime packet waiting time.

3.1.1 Comparison of Various Multiplexing and Tunneling Techniques

In this scenario only voice data is assumed and the multiplexing timer is set to 0.5 msec. At higher link rates more connections must be active to reach the critical load. Due to the economy of scale, the bandwidth can be better exploited without violating the QoS requirements of the data. In addition, at higher bandwidth more voice samples can be multiplexed into an IP-packet, which reduces the header overhead, while the

for tunneling protocols the header overhead remains constant (Figure 7). So, the proportion of user data transported on the link is higher for multiplexing protocols.

The size of the IP/UDP/RTP protocol for multiplexing does not really matter, the overhead for Mux-3-12 and Mux-3-0 hardly differs while the size of the multiplexing header is crucial: Mux-13-12 shows a substantially higher overhead.

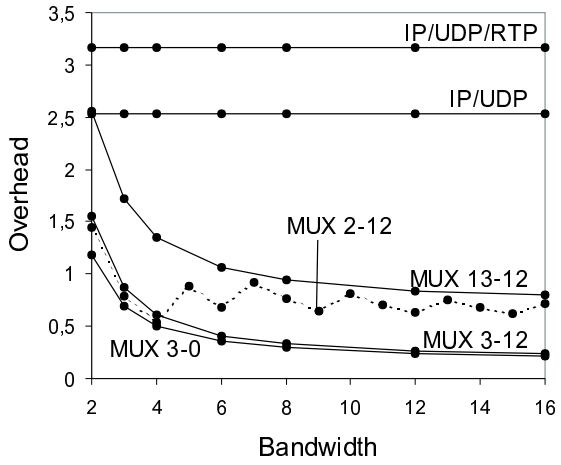

Figure 7: Multiplexing decreases the protocol overhead.

For the multiplexing protocol with only one byte reserved for the CID value (Mux-2-12) and n customers in the system, $n_c = \lceil \frac{n}{256} \rceil$ RTP connections must be opened and spaced on a common flow basis. To take that into account, we multiply the RTP/UDP/IP header size by n_c. The 2 byte CID alternative outperforms the 1 byte CID version at a bandwidth of more than 4 Mbps. Therefore, Mux-3-12 is used for the following studies.

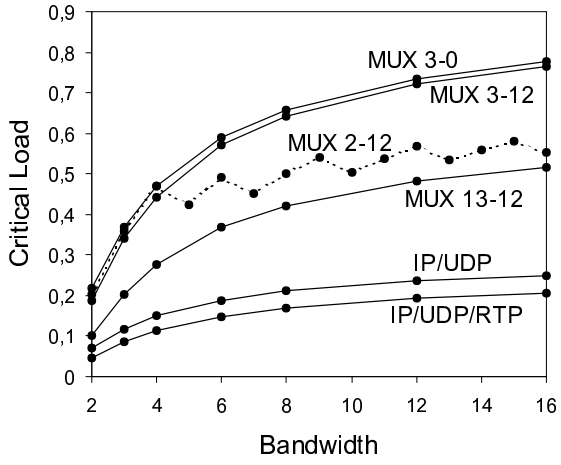

Figure 8: Multiplexing allows higher link utilization by user data.

Although the overall link utilization is fairly similar for all tested protocols, the link utilization by user data differs substatially according to the proportion of user data in a sent IP-packet (Figure 8). At 16 Mbps the light weight multiplexing protocols (Mux-2-12, Mux-3-12, and Mux-3-0) clearly outperform tunneling: the tunneling overhead is 13 times larger and, therefore, the link exploitation for user data is

260% higher . In other words, to carry 450 voice calls, 16 Mbps are needed for RTP/UDP/IP tunneling while for multiplexing (Mux-3-12) 6 Mbps are needed.

3.1.2 Optimum Packet Size for CS Data

In this scenario only CS data traffic is assumed, the bandwidth is 8 Mbps and the timer value is set to 0.5 msec. A CS data source has a bandwidth of 64 kbps, i.e. 1 byte per 125 μsec. The optimum packet size is about 16 bytes for the transportation in the fixed network part of UMTS, however values from 8 to 50 bytes yield also good link utilizations by user data (Figure 9). The existence of an optimum is due to a tradeoff between overhead and variablility in the resulting IP packets. If the packet size is small, the multiplexing header is relatively large and the proportion of transported user data is small. If the packet size is large, the coefficient of variation of the resulting IP packet stream is relatively high, leading to lower link utilization due to higher waiting times at the spacer.

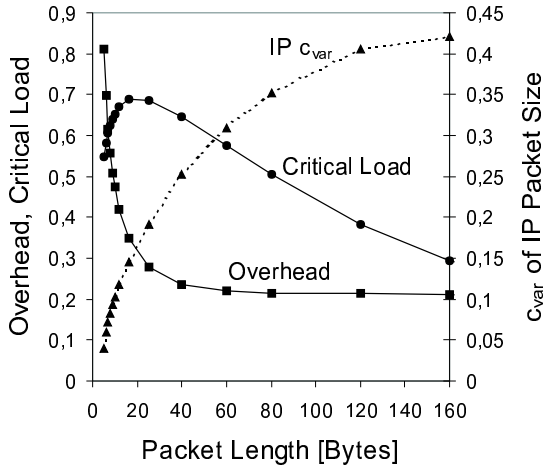

Figure 9: The optimum CS data packet length

The reason for the reduced variance in IP-packet size lies in the geometric interarrival time of the CS data packets. Short packets evoke a high arrival rate reducing the variance of the distribution. This makes sense in the real world, too, since every IP packet has a packet of each connection if the packet size is only 4 bytes and the multiplexing timer is 5 msec. Then the packet size is deterministic and has a coefficient of variation of 0 which is the limit in Figure 9, too. This is not an effect of the discrete nature and the extreme range for the geometric distribution. The same phenomenon is obtained for its continuous time equivalent, the exponential distribution, too. The coefficient of variation for the number of packet arrivals in an interval of length t is $c_{var} = \frac{1}{\sqrt{\lambda \cdot t}}$ [19] where λ is the packet arrival rate. For a packet size of 5 bytes, the arrival rate is λ =112.5 msec^{-1} and the interval length is the value for the multiplexing timer t=0.5 msec. This results in a coefficient of variation of 0.13.

3.1.3 Impact of the Voice Proportion

In a real world application there is a mixture of voice traffic and CS data traffic. Therefore, it is of interest to know the system behavior depending on the ratio of the different traffic types. A voice proportion of p means that p of the total user data originates from voice traffic. Figure 10 shows that to compute the critical load for a certain voice proportion, the critical load for the CS data and voice carrying systems can be almost linearly interpolated. For a CS data packet size of 40 bytes, the voice proportion has hardly any impact. If the CS data packet length is very large, the overall performance of the system degrades rapidly with increasing CS data proportion. A means to alleviate this is to relax the delay requirement for CS data and carry them with lower priority.

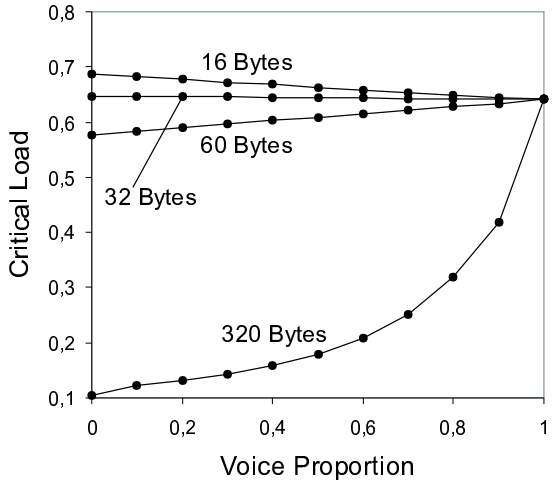

Figure 10: Critical load depends on voice proportion and CS data packet length.

3.1.4 Optimum Value for the Multiplexing Timer for CS data

In [11] was shown that the variance of IP-packet sizes has an influence on the optimum value of the multiplexing timer. If this variance is higher, so is also the variance of the spacing time. Then, the quantile for the delay budget is also higher and so the multiplexing timer has to be shorter to meet the QoS criterion. Since the CS data packet size also impacts the coefficient of variation of the IP-packet size, the optimum timer value depends on it, too. In addition, the fact that smaller CS data packet sizes induce more overhead also plays a role. Figure 11 shows that the optimum timer for 40 byte CS data packets is about 0.5 msec, which is the same as for voice traffic. For small packet sizes, the timer should be large, for large packet sizes vice versa. A timer value of 0.5 msec seems to be a good compromise for all packet sizes.

3.2 Leaky Bucket Traffic Contracts

Since no spacing is performed, only the multiplexing time contributes to the realtime packet waiting time. The delay budget QoS criterion can always be met if the multiplexing timer is set to the value of the delay budget (1 msec). That means, that twice as many packets can be multiplexed making multiplexing even more effective, especially at lower bandwidth. The bucket size is the amount of transfer capacity that can be temporarily borrowed from the network. From the network point of view, this is costly and consequently larger bucket size parameters must be billed higher. Therefore, the bucket size as well as the bandwidth need to be minimized to realize a cost-effective data transmission.

3.2.1 Control Function of the Bucket Size Parameter

A given bandwidth, say 8 Mbps, can be exploited to a different extent depending on the bucket size parameter. Conversely, to obtain a certain link utilization by user data, a different bucket size can be set. The required bucket size depends heavily on the bandwidth: At higher bandwidth, more connections can be supported and within the multiplexing time TCU more realtime packets are collected into an IP-packet. This has to be accommodated in the virtual spacer. Therefore, the notion of the normalized bucket size is introduced which is the bucket size divided by the bandwidth. It denotes the maximum delay that would be caused by a spacer.

A higher critical load requires a larger bucket size. The fact that the normalized bucket size is smaller for higher bandwidth is due to the economy of scale for leaky bucket contracts (Figure 12). The overall link utilization ρ is higher for small bandwidth since more overhead is carried to yield the same user data utilization. Economy of scale makes a higher link exploitation possible while using a tolerable normalized bucket size.

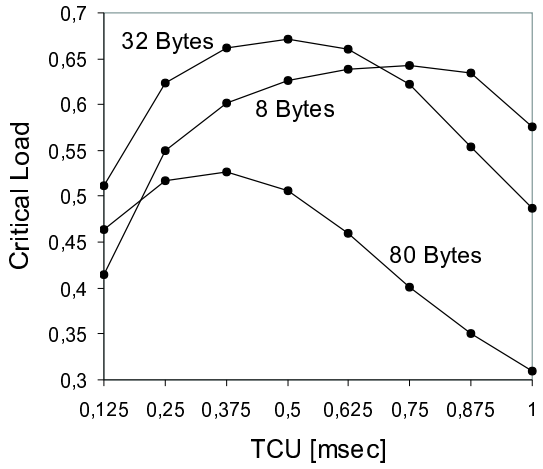

Figure 11: The optimum multiplexing timer depends on the CS data packet size.

3.2.2 The Tradeoff between Bandwidth and Bucket Size

So far, the considerations were of more general nature. Now, the support of 250, 500, and 750 over a network with leaky bucket traffic contract is considered. Figure 13 shows the tradeoff between bandwidth and the required bucket size. At a higher bandwidth a smaller bucket size is required to carry the same amount of traffic. Keeping the amount of traffic fixed, the network is less loaded at higher bandwidth and, therefore, the queue in the virtual spacer is shorter. The bucket size can not fall below a certain threshold that is 920, 1344, and 1716, respectively. At those points, the IP-packet size distribution has a quantile of 10^{-6} which is the minimum loss probability. If the bucket size was smaller, the probability to discard an IP-packet in spite of an empty virtual spacer would exceed 10^{-6}. With multiplexing and a leaky bucket traffic contract, 250 connections can be supported by a bandwidth of 3 Mbps while for tunneling and peak rate spacing, more than 10 Mbps are needed. The tradeoff between bandwidth and bucket size gives multiple possibilities to dimension parameters for a leased line. When tariffs come into play, the least cost parameters can be found using a cost function that must take both bandwidth and bucket size into account.

3.2.3 Optimum Packet Size for CS Data

We choose a link of 8 Mbps and set the bucket size parameter to 1600 bytes. The optimum packet size maximizes the link utilization in this given scenario. The reason for that phenomenon is again the tradeoff between overhead and variance again. As for peak rate spacing, the best packet length for CS data is 16 bytes. But note that the curve in Figure 14 is different from the one in Figure 9. The header overhead shows even a minimum at 60 bytes CS data packet length. On the one side, the overhead reduces with larger packet sizes since less multiplexing headers need to be carried but on the other side, the critical load shrinks and less data can be transmitted, i.e., less data can share the fixed IP/UDP/RTP protocol overhead.

3.2.4 The Impact of the Voice Proportion

As before, the bandwidth is set to 8 Mbps and the bucket size to 1600 bytes. The influence of the voice proportion of the traffic is similar to the one with peak rate spacing. The critical load decreases or increases almost linearly with increasing voice proportion depending on the CS data packet size. For a CS data packet size of 32 bytes, the voice proportion has hardly any effect on the critical load of the system.

4 Conclusion

We established a model for the transmission of compressed voice samples and 64 kbps CS data in UMTS using RTP multiplexing and showed different performance tradeoffs illustrated by analytical results.

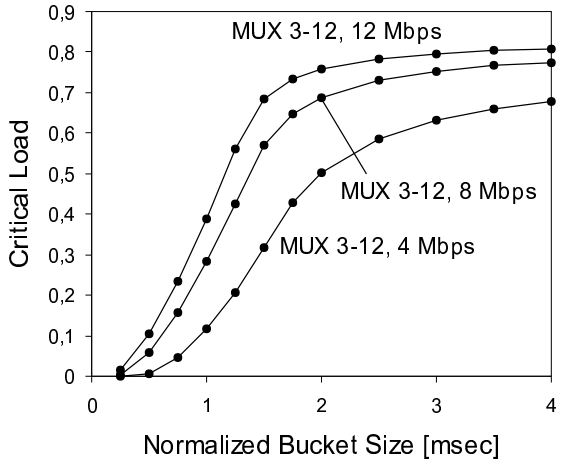

Figure 12: The normalized bucket size depends on critical load and bandwidth.

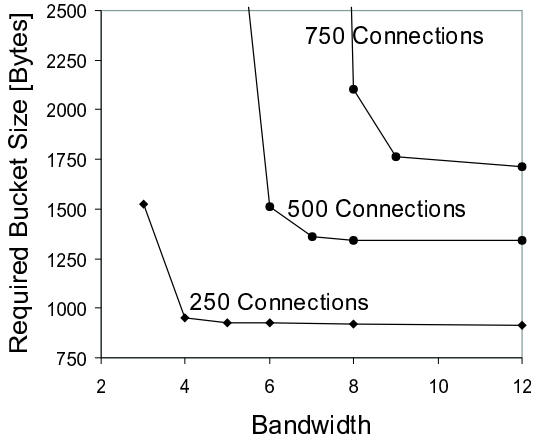

Figure 13: The tradeoff between bandwidth and bucket size

Multiplexing voice data increases the utilization of the network for user data considerably. However, it is crucial to have the CID value in the multiplexing protocol large enough. If circuit switched data are sent over the same network, their packet size highly influences on the performance of the system. Hence, to optimize the data transport in the wireline part of UMTS, the definition of the circuit switched data services must take the influence of different packet sizes into account.

For a system that applies peak rate spacing, the performance of a multiplexing system can be optimized by setting an appropriate value for the multiplexing timer. A network contract requiring a leaky bucket size description

Having a VBR traffic contract, policing using leaky bucket parameters is performed at the ingress of the network. With VBR, the booked bandwidth can be better exploited and economy of scale for the bucket size is observed with increasing bandwidth. The dimensioning of leaky bucket parameters for a multiplexing system shows a tradeoff between bandwidth and leaky bucket size. This can be used to minimize transmission costs. The optimum CS data packet length is the same as for CBR and for a packet size of 40 bytes the voice proportion in the offered load does not matter.

Further work will investigate the transport of voice and CS data with different priorities assuming that CS data has a relaxed requirement on packet delay.

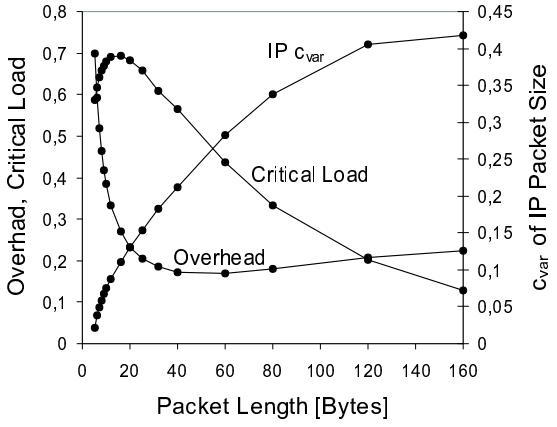

Figure 14: The optimum CS data packet length

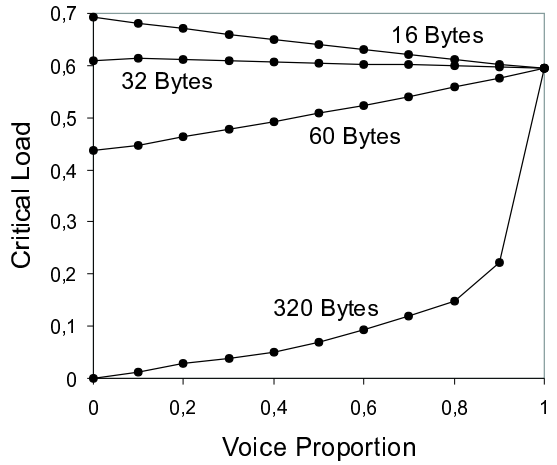

Figure 15: Critical load depends on voice proportion and CS data packet length.

References

[1] 3GPP, "3G TR23.922 version 1.0.0: Architecture for an all IP network," Oct. 1999.

[2] K. El-Khathib, G. Luo, G. Bochmann, and F. Pinjiang, "Multiplexing scheme for RTP flows between access routers <draft-ietf-avt-multiplexing-rtp-00.txt>." http://www.ietf.org/internet-drafts/draft-ietf-avt-multiplexing-rtp-01.txt, Oct. 1999.

[3] N. Gerlich and M. Ritter, "Carrying CDMA traffic over ATM using AAL–2: A performance study," Technical Report, No. 188, University of Wuerzburg, Institute of Computer Science, Sep. 1997.

[4] N. Gerlich and M. Menth, "The performance of AAL-2 carrying CDMA voice traffic," in *11th ITC Specialist Seminar*, (Yokohama, Japan), Oct. 1998.

[5] N. Gerlich, *Transporting Wireless Network Traffic on Wired Networks - A Performance Study*. PhD thesis, University of Wuerzburg, Faculty of Computer Science, Am Hubland, Apr. 1999.

[6] M. Menth and N. Gerlich, "A numerical framework for solving discrete finite markov models applied

to the AAL-2 protocol," in *MMB '99, 10th GI/ITG Special Interest Conference*, (Trier), pp. 0163–0172, Sep. 1999.

[7] B. Subbiah, S. Dixit, and N. R. Center, "Low-bit-rate voice and telephony over atm in cellular/mobile networks," *IEEE Personal Communications*, pp. 37–43, Dec 1999.

[8] S. Blake, D. Black, M. Carlson, S. Davies, Z. Wang, and W. Weiss, "RFC2475: An architecture for differentiated services." ftp://ftp.isi.edu/in-notes/rfc2475.txt, Dec. 1998.

[9] J. Wroclawski, "RFC2210: The use of RSVP with IETF integrated services." ftp://ftp.isi.edu/in-notes/rfc2210.txt, Sep. 1997.

[10] S. Shenker, C. Partridge, and R. Gueria, "RFC2212: Specification of guaranteed quality of service." ftp://ftp.isi.edu/in-notes/rfc2212.txt, Sep. 1997.

[11] M. Menth, "Carrying wireless traffic in UMTS over IP using realtime transfer protocol multiplexing," in *12th ITC Specialist Seminar*, (Lillehammer, Norway), pp. 13 – 25, March 2000.

[12] 3GPP, "TSGR3#7(99)c05: Study item (ARC/3) overall delay budget within the access stratum." status report, Sep. 1999.

[13] S. Deering and R. Hinden, "RFC2460: Internet protocol version 6 (IPv6) specification." ftp://ftp.isi.edu/in-notes/rfc2460.txt, Dec. 1998.

[14] J. Postel, "RFC768: User datagram protocol." http://www.ietf.org/rfc/rfc0791.txt, Sep. 1980.

[15] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RFC1889: RTP - a transport protocol for real-time applications." ftp://ftp.isi.edu/in-notes/rfc1889.txt, Jan. 1996.

[16] M. Degermark, B. Norgren, and S. Pink, "RFC2507: IP header compression." ftp://ftp.isi.edu/in-notes/rfc2507.txt, Feb. 1999.

[17] S. Casner and V. Jacobson, "RFC2508: Compressing IP/UDP/RTP headers for low-speed serial links." ftp://ftp.isi.edu/in-notes/rfc2508.txt, Feb. 1999.

[18] The ATM Forum, *Traffic Management Specification, Version 4.0*, Apr. 1996.

[19] P. Tran-Gia, *Analytische Leistungsbewertung verteilter Systeme*. Berlin, Heidelberg, New York: Springer, 1. ed., 1996.