In Proc. of International Symposium on Telecommunications (IST2005), Shiraz, Iran, September 2005

Adaptive Coding and Packet Rates for TCP-Friendly VoIP Flows

C. Mahlo, C. Hoene, A. Rostami, A. Wolisz
Technical University of Berlin, TKN, Sekr. FT 5-2 Einsteinufer 25, 10587 Berlin, Germany.
Emails: mahlo@cs.tu-berlin.de, hoene|rostami|wolisz@tkn.tu-berlin.de

Abstract

If low-rate VoIP is transmitted over congested links, both the coding and the packet rate need to be adapted to achieve the best conversational quality: On an Ethernet network it is better to jointly change packet rate and speech coding rate; in WLAN networks it is sufficient to change the packet rate. We also study whether TCP-friendly rate control algorithms are suitable for dynamically adapting packet and coding rate of a VoIP flow. Our results show that cannot be used for VoIP applications as they only control the packet rate do not consider packet sizes. Also, they do not transmit small (speech) packets fairly. We present an enhanced version of the TCP friendly rate control (TFRC) protocol that transmits small packets fairly and evaluate it with network simulations.

Keywords: VoIP, packetisation, packet rate, coding rate, TCP-friendliness, TFRC, packet switching overhead.

Introduction

Voice over IP (VoIP) is increasingly considered to offer telephone services because it can be used over existing IP-based access networks and backbones. In opposite to TCP-based traffic, which still accounts for majority of data transport in IP networks, the sending rate of VoIP flows usually remains constant over the duration of a call. Of course, available bandwidth is often plenty enough for VoIP so that there might not be an immediate need to adapt the sending rate of VoIP flows dynamically. However, existing and upcoming wireless networks like cellular networks, sensor networks, or multiple hop ad-hoc networks might require an efficient and low-rate transmission of application data because wireless links on those networks have sparse capacities. This fact is partly due to the tremendous costs of the communication infrastructure (e.g. base stations), partly due to the tight energy consumption constrains of battery powered nodes, but also because of restrictions on wireless spectrum usage.

In this paper we discuss how such an efficient communication may be achieved for voice over IP (VoIP) applications. Our approach is to dynamically adapt both the coding rate and packetisation in an TCP friendly manner in order to optimize the quality of VoIP flows.

The conversational call quality of VoIP flows depends on the transmission delay and on the speech quality [2]. For example, a higher coding rate usually gives a better speech quality. On the other hand, if the packet rate is set low and the packetisation time is high, the overall transmission delay increases and conversation call quality is harmed.

Both packet and coding rate influence the gross bandwidth requirements of VoIP, either by the amount of user data or the number of packet headers. This leads to an interesting question: If the link is congested and packet losses occur, should we adapt the coding or the packet rate? As we show in this paper, the underlying physical link technology defines the best adaptation strategy.

As a second contribution we consider how to use TCP-friendly congestion control to adapt a low-rate VoIP flow dynamically (Fig. 1).

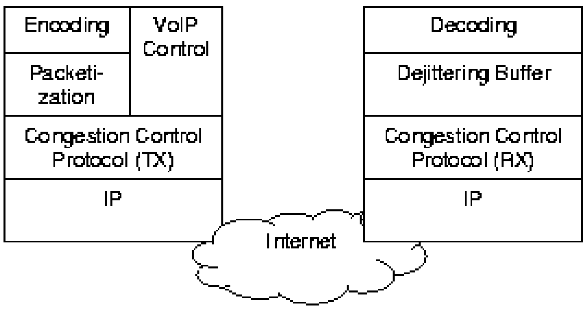

Fig. 1: TCP-friendly VoIP control.

Internet's congestion control algorithms regulate the resource distribution among multiple flows and prevent a congestion collapse of the Internet [4]. Congestion control in the Internet is based on TCP or TCP-friendly transport protocols.

It is known that the throughput of a TCP-like protocol scales linearly with the packet size [5]. Thus, the optimal packet size for TCP-friendly protocols is the maximal transfer unit (MTU). However, this causes some problems in the case of Internet telephony. In fact, due to the interactive nature of telephony, it would be more useful if small packets at a higher rate are generated instead of generating large packets at a lower rate. The reason is that smaller packets enjoy lower packetisation time and thus lower transmission delays.

Our study shows that a fair treatment of small packets depends on packet switching overhead of the bottleneck link. As a consequence, we show that if the packet switching overhead would be known to the congestion control protocol, the protocol could act fair.

The remainder of this paper is organized as follows. First, we continue with a related work section. In chapter two we present the consideration on coding and packet rate. Then, we discuss the difficulty of dealing with small packets TCP-friendly. Finally, we conclude.

1. Related Work

The requirements of VoIP differ greatly to those of TCP-based applications: First VoIP is delay sensitive. An increased delay harms the conversational call quality. Second, to cope with high delay variations a dejittering buffer is required, which increases the end-to-end delay. Last, VoIP suffer from packet loss. Packet loss decreases the speech quality. An overview on VoIP quality requirements is given in [2].

Bolot [12] has suggested to control the transmission of packet audio by a joined rate and forward error control (FEC). Based on the receivers' packet loss feedback, the sender lowers coding rate and/or increased the amount of FEC. Bolot demonstrated the effeteness of the propose solution experimentally by transmitting multiple sources of a common 100 kb/s link.

In [13] the authors simulated the transmission of voice communications over IP networks adapting the coding rate. They showed that an adaptive coding rate outperforms a fixed rate approach, if transmitted parallel to over TCP-like flows and even constant-rate flows. As Bolot, the authors did not apply a TCP friendly control schemes nor did they alter the packet rate.

Boutremans et al. [14] optimized the rate-, error- and delay control for interactive VoIP. By applying a quality model, which takes into account both packet loss and delay impairment, the author could find analytically the best coding rate and amount of FEC for a given packet loss process and round trip time. Also, they included TCP friendliness module, which control the application data rate, keeping the packet rate constant.

In a Voice over WLAN system Veeraraghavan et al. [15] considered an adaptive packetisation. The proposed system support a single-hop and none IP based transmission of interactive speech only.

None of the rate-control protocols for VoIP jointly optimize the packet and the coding rate in a TCP-friendly manner.

Widmer presents a comprehensive survey on the latest TCP-friendly congestion control protocols [6] and lists eleven proposals. In [1] we evaluated different rate control algorithms. Our simulation results indicated that TCP-friendly rate control (TFRC [7]) is most suitable for VoIP, mainly due to its smooth sending rate. Thus, in this paper we concentrate on TFRC.

TFRC [7] is based on analytic derived equation, which calculates the packet sending rate of a TCP flow. To gather the parameters necessary for the TCP equation, the receiver sends back loss rate, round trip times and round trip time variation. The throughput of TCP is largely influenced by the parameters round-trip time t_{RTT} retransmission timeout value t_{RTO}, segment size s, and the packet loss rate p. The loss rate is measured in terms of loss intervals, spanning the number of consecutive loss intervals. A certain number of loss intervals are averaged, using decaying weights so that old loss intervals contribute less to the average timeouts. The following equation calculates the sending rate T of a TCP connection.

$$T = \frac{s}{t_{RTT}\sqrt{\frac{2p}{3}} + t_{RTO}\sqrt{\frac{27p}{8}}\left(p + 32p^3\right)} \tag{1}$$

In [8] the authors suggest a technique for router which detects congestion early to avoid extensive packet loss in a network. The Random Early Detection (RED) algorithm monitors the queue lengths of routers. With increasing queue length the drop probability is increased. In contrast, drop-tail queues just drop packets when the queue is full. By early packet losses RED indicates the TCP protocol that congestion will probably occur in the future. Thus, the TCP protocol drops its sending rate to prevent congestion. RED works with two different modes. The first counts the number of packets in the queue and the second counts the number of bytes in the queue.

In [5] the negative impact of a variable packet size on the TFRC protocol is analyzed. To recover fairness between flows with different packet sizes a pure network-based approach using a RED gateway in byte mode does not ensure fairness. The authors argue that the TCP equation cannot be used in conjunction with variable packets and a simple modification by using the MTU as packet size is insufficient. The authors present four new algorithms and conclude that the knowledge on the correlation between loss rate and packet size is important.

2. Coding and Packet Rate

Both packetisation and coding rate influence the gross bandwidth requirements of VoIP. The coding rate depends on the operational mode of codec. For example, AMR supports eight different modes between 4.75 and 12.2 kbps [3]. Higher coding rates have a better speech quality. The packetisation controls how many speech frames are concatenated in one VoIP packet. For example, AMR generated a speech frame every 20 ms and VoIP packets have a lengths of 20 or 40 ms. If a high packetisation is selected, the packet rate is low and less packet headers are sent.

Let us study, how a VoIP flow shall be adapted to a congested link that was limited bandwidth. We assume that the capacity of a connection remains constant and is known. The transmission delay is given and remains constant for each packet. The question to answer is how to choose the optimal coding rate and packetisation under these circumstances. In [2], we discuss this question for circuit-switched and Ethernet links: We calculate the best VoIP configuration[1] on an Ethernet link if both packet and coding rate are ideally chosen to limited bandwidth. We assume the AMR codec and a 150 ms end-to-end delay excluding the packetisation time. The Fig. 2 shows that the perfect packetisation increases if the available bandwidth drops. Only at a very low bandwidth the coding rate needs to be decreased, too.

Let us extend these studies and analyze the IEEE 802.11b WLAN protocol. Due to compatibility reasons 802.11b is the MAC protocol, which seems to have the highest packet switching overhead of all common MAC protocols. The overhead for each packet has been calculated and validated in [10]. We assume a long physical preamble and no RTS/CTS. Then, an 11 Mbps WLAN packet has an overhead of about 1500 bytes (For the Ethernet link we used a packet overhead of about 67 bytes.) As in Fig 2, Figure 3 displays the optimal configuration of packet and coding rate, if the bandwidth is limited. Interestingly, the coding rate needs not to be lower over a width range of networking conditions. On the packet rate has to be decreased.

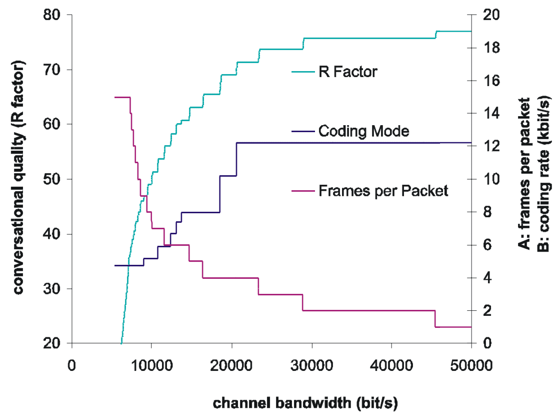

Fig. 2: Choosing optimal coding rate and packetisation on Ethernet-like packet-switched link.

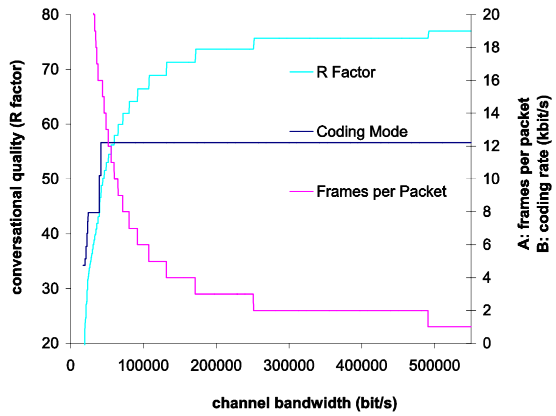

Fig. 3: Choosing optimal coding rate and packetisation on an IEEE 802.11b WLAN link.

In the case of limited bandwidth, the rate parameter to adapt depends on the underlying networking technology. In the case of a circuit-switched link without any packet overhead, the coding rate should be lowered [2]. On an Ethernet-like link, both coding and packet rate have to be adapted to achieve an optimal quality. On a link technology with large packet switching overhead like IEEE 802.11b, the packet rate should be lowered but the coding rate can remain the same. Thus, this leads to an interesting conclusion on how to support congestion control for low-rate VoIP. If the bandwidth is to low for even narrow-band telephony, instead of changing the just coding rate both the packet and the coding rate has to be reduced.

---

[1]To measure the conversational call quality we apply the ITU E-model [2]: The R factor considers both speech quality and delay. Zero refers to bad quality; 100 is perfect call quality.

3. Small Packets

The TCP-friendly throughput scales linear with the packet size as shown in [5]. Thus, in cases of congestion TCP transmits packets with the largest packet size (MTU) to achieve the highest throughput. In case of telephony this leads to a problem. Because of the interactive requirement it would be useful, if small packets at a higher rate are generated instead of large packets at a low rate. Smaller packets have a low packetisation time and thus decrease the round-trip time.

Flows of small packets are discriminated because if a flow decides to transmit small packets, the link is less utilized. Then, other parallel flows increase their sending rate to fill up the capacity gap. Thus, the parallel flows benefit. This fact has been shown – for example – with ns-2 simulations [1]. This behavior is not fair because the bandwidth distribution should be independent of the packet size.

How should be TCP-friendliness be modified so that the flow with small packet does not suffer to that extend that the other flows gain a benefit? Widmer suggested a definition of TCP-friendliness [6], which is: *A flow is TCP-friendly, if other flows devote the same amount of resource as if the flow would be a TCP flow.*

In general, TCP-friendly rate control algorithms are not suitable for the transmission of voice flows that need to transmit small packets. For this reason we suggest two alternate solutions which are suitable to improve the fairness.

First, small packets can be treated differently in the router, e.g. by the Random Early Dropping. The idea was to use RED in byte mode instead of packet. In this case the arrival of a large packet causes the queue to fill faster and the probability increases that the large packet will be dropped. On the other side small packets will be dropped with a lower probability. Thus, TCP-friendly flows with small packets are transmitted at higher rates because of a lower loss rate.

Second, the TCP-friendly rate protocols like TFRC can be changed so that data streams with small packets are not disadvantaged. The idea was to change the TFRC sender, which calculates the maximal sending rate. If a TFRC sender uses a small package size s for the data transmission, this upper limit of the possible sending rate is reduced. We propose therefore an empirical extension of the TCP equation, which produces fairness between different package sizes. The calculated sending rate is enhanced by a weighted ratio of the maximum and the current package size:

$$T' = T \cdot factor \qquad (2)$$

$$factor = \left(\frac{MSS}{s}\right)^{\alpha} \quad with \quad 0 < \alpha \leq 1$$

T has the unit of byte/s and is a function of the packet size s. MSS describes the maximum size of a data packet. The difficulty of the adaptation is in the suitable choice of the weighting exponent α, which depends on the properties of the used transmission medium. For example, Fig. 4 shows the graphical course of the factor for selected values of α and a typical value of MSS = 1500 byte. This equation works well for packet size about 120 bytes, as simulation have shown [1].

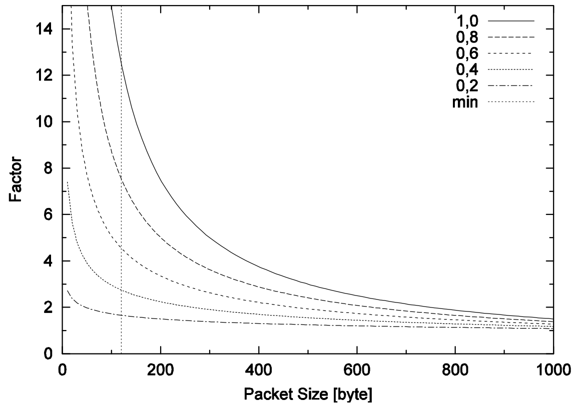

Fig. 4: Adaptation factor α

4. Simulation Results

The evaluations of the proposals were performed with the network simulator ns-2 version 2.1b9 [11]. The scenario is shown in Fig. 5 and contains a mixture of wired and wireless network. The bottleneck here is created by the shared wireless medium. On the wireless channel we used a Gilbert/Elliott bit error model to simulate transmission errors.

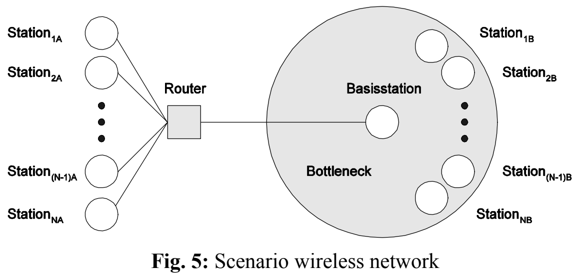

Fig. 5: Scenario wireless network

For the evaluation of the protocols we chose a wireless transmission rate of 2MBit/s and an activated RTS/CTS-exchange. Within these simulations we could take into account the influences of the wireless channel on the behavior of the protocols. The available bandwidth of the wireless medium was exhausted by five simultaneous TFRC connections. The senders were placed in the wired network and the receivers accordingly in the wireless access network. The simulation time was 120s. The common, wired network path was configured with a transmission rate of 4MBit/s in order not to be the transmission bottleneck. The delay of this link was 50ms and had a drop tail queue with a sufficient packet buffer size. The wired nodes were connected by 10MBit/s links with different but constant delays.

In the first simulation, we simulated five parallel TFRC flows sending with packet size of 600 bytes using a drop-tail queue. We display the sending rate over time in Fig. 6. All five flows have a similar sending rate. In Fig. 7 one flow sends with small packets bidirectional. One can clearly see that the sending rate of this flow is far lower whereas the other four flows with large packets gain more bandwidth.

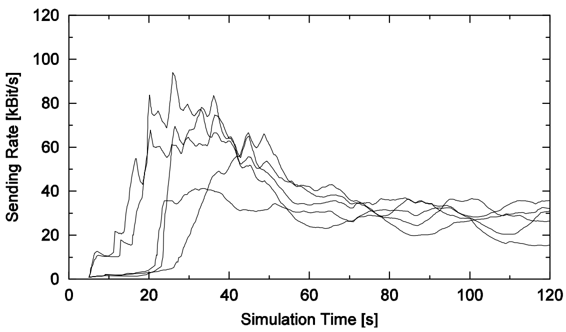

Fig. 6: All five TFRC flows use 600 byte packets.

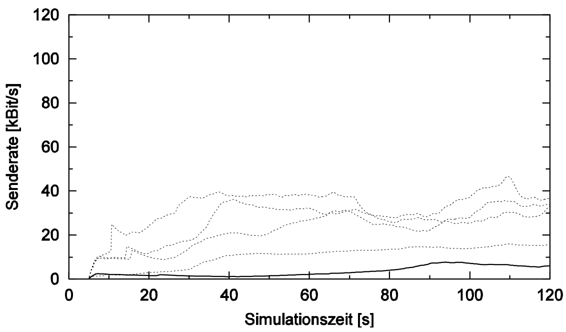

Fig. 7: Four TFRC flows use 600 byte packets and one (bold line) uses 159 byte packets.

Next, we simulated the RED queue in byte-dropping mode. In comparison to the drop tail queue results the variance of the sending rate is higher (Fig. 8). The TFRC protocol seems to dislike REDs random packet drops. However, all flows have a similar throughput. Thus, small packets are treated fairly. RED in the byte-mode increases fairness for small packets at the expense of stability.

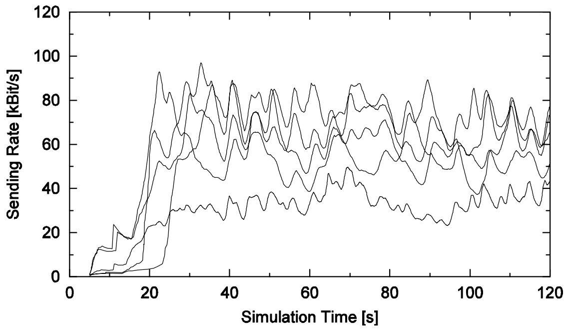

Fig. 8: TFRC and RED queue in byte mode

Figure 9 shows the results of our simulations with the improved TFRC protocol. Flows with small packets are treated fairly and have a stable sending rate. However, the factor α has to be chosen properly. By empirical examination and detailed simulations [1] we have determined a suitable value of α=0.6 for the transmission within the wireless Ethernet.

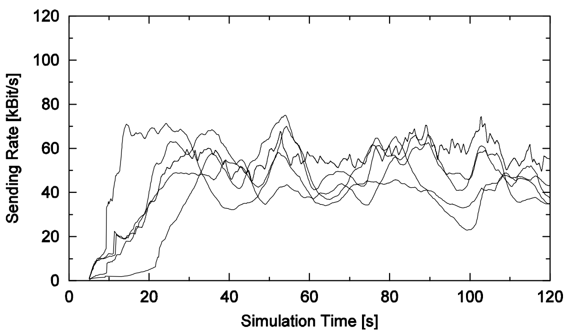

Fig. 9: TFRC improved

In other simulations [1] we replace the wireless link by a shared Ethernet link. Then the factor α is lower due to the lower MAC-overhead and medium access by collision detection. We have found out a suitable value of α=0.35 for this case.

Conclusion

If a low-rate VoIP flow needs to reduce its bandwidth to prevent congestion, it must reduce both coding and packet rate. On WLAN links it is sufficient to lower the packet rate as they have a large packet switching overhead. This packet switching overhead includes the packet header but also MAC packet header, contention period, physical framing, and collision[2].

[2]Interestingly, using IP header compression does not decrease the packet switching overhead significant, as it only compresses IP, UDP, and RTP header, which contribute only marginally to the switching overhead.

If one wants to use TCP-friendly rate control to select the best coding and packet rate of a VoIP flow, some fundamental properties of TCP friendliness have to consider:

- TCP and TFRC just control the packet rate and do not consider the bandwidth or the packet sizes. They assume that every flow transmits with the maximal transfer unit.
- A flow shall be called TCP-friendly, if other flows devote the same amount of resource as if the flow would be a TCP flow.
- To overcome with the issue of small packets, using Random Early Dropping (RED) in the byte-mode has been proposed. But RED controlled flows are not stable and RED is not implemented in all routers.
- Alternative, we propose an enhanced TFRC, which considers the packet switching overhead of the bottleneck path to share the link fairly among flows with different packet sizes.

We think that VoIP, video, games, and transactions like communication can benefit from small packets. Nevertheless, no standard proposal has been made on how to support small or variable size packets TCP-friendly. Also, it would be required to inform the sender about the packet switching overhead of the bottleneck link. Otherwise, a fair treatment of flows with small packets is not possible.

References

[1] C. Mahlo, “Congestion control for low-rate interactive speech,” Master’s thesis, Technical University of Berlin, TKN, Germany, Feb. 2003.

[2] C. Hoene, H. Karl, and A. Wolisz, “A perceptual quality model for adaptive VoIP applications,” in Proc. of SPECTS’04, San Jose, CA, July 2004.

[3] E. T. S. Institute, “Universal Mobile Telecommunications System (UMTS), AMR Speech Codec, General Description,” 3GPP TS 26.071 Version 5.0.0 Release 5, June 2002.

[4] S. Floyd and K. Fall, “Promoting the use of end-to-end congestion control in the internet,” *IEEE/ACM Trans. Netw.*, vol. 7, no. 4, pp. 458–472, 1999.

[5] J. Widmer, C. Boutremans, and J.-Y. L. Boudec, “End-to-end congestion control for TCP-friendly flows with variable packet size,” SIGCOMM Comput. Commun. Rev., vol. 34, no. 2, pp. 137–151, 2004.

[6] J. Widmer, R. Denda, and M. Mauve, “A Survey on TCP-Friendly Congestion Control,” IEEE Network, vol. 15, no. 3, pp. 28– 37, May 2001.

[7] S. Floyd, M. Handley, J. Padhye, and J. Widmer, “Equation-based congestion control for unicast applications,” in SIGCOMM 2000, Stockholm, Sweden, Aug. 2000, pp. 43–56.

[8] S. Floyd and V. Jacobson, “Random Early Detection Gateways for Congestion Avoidance,” *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, Aug. 1993.

[9] ITU, “The E-Model, a computational model for use in transmission planning,” ITU-T Recommendation G.107, May 2000.

[10] M. Arranz, R. Aguero, L. Munoz, and P. Mahonen, “Behavior of UDP-based applications over IEEE 802.11 wireless networks,” in 12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, vol. 2, Sept. 2001, pp. 72–77.

[11] “The Network Simulator -ns-2.” [Online]. Available: http://www.isi.edu/nsnam/ns

[12] J.-C. Bolot, S. Fosse-Parisis, and D. F. Towsley, “Adaptive FEC-based error control for internet telephony,” in Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’99), vol. 3, New York, NY, Apr. 1999, pp. 1453–1460.

[13] A. Barberis, C. Casetti, J. C. De Martin, and M. Meo, “A simulation study of adaptive voice communications on IP networks,” *Computer Communications*, vol. 24, no. 9, pp. 757–767, May 2001.

[14] C. Boutremans and J.-Y. L. Boudec, “Adaptive joint playout buffer and FEC adjustment for internet telephony,” in Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003), vol. 1, San-Francisco, CA, Apr. 2003, pp. 652–662.

[15] M. Veeraraghavan, N. Cocker, and T. Moors, “Support of voice services in IEEE 802.11 wireless LANs,” in Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2001), vol. 1, Los Alamitos, CA, 2001, pp. 488–497.