Service Differentiation Extensions for Elastic and Real-Time traffic in 802.11 Wireless LAN

Albert Banchs[a], Xavier Pérez[a], Markus Radimirsch[b], Heinrich J. Stüttgen[a]

[a] C&C Research Laboratories, NEC Europe Ltd., Heidelberg, Germany

[b] Institut für Allgemeine Nachrichtentechnik, University Hannover, Germany

Abstract—

QoS in wireless networks has a special relevance due to the scarce bandwidth available in such networks. This contribution addresses this issue by extending the MAC protocol of the IEEE 802.11 wireless LAN standard. The extension is divided in two steps. In the first step, real-time traffic is distinguished from elastic traffic by a priority scheduling approach in order to meet the delay requirements of e.g. voice communication. In the second step, service differentiation is introduced for elastic traffic, based on a relative differentiation model. In this model, a high priority service always receives a higher throughput than a low priority one. The proposed architecture has been validated via simulation. Results for real-time traffic show that the proposed approach leads to delays sufficiently low if admission control is properly applied. Elastic traffic achieves the desired differentiation in all simulated scenarios.

I. Introduction

The world of data communication has undergone many changes over the last few years. Probably the most important one is the convergence of voice, video and data communication under the roof of the Internet Protocol (IP) suite. Originally, IP was designed to support elastic services, i.e. data applications like file transfer, electronic mail and remote terminal. Elastic services are tolerant of delays and, even though they benefit from increasing data rates in terms of user satisfaction, still work at low data rates. Voice services, in contrast, require a certain minimum rate and suffer significantly from high delay and delay variation.

In the last years, considerable effort has been made to provide QoS to wired networks with two principal proposals: Integrated Services and Differentiated Services. While Integrated Services [1] provides hard QoS guarantees at the cost of having complex and sophisticated mechanisms and protocols, Differentiated Services [2], increasingly popular, requires much less control and signaling, scaling better to large networks but providing softer QoS guarantees.

Both Integrated and Differentiated Services are based on *absolute performance levels.* These architectures are based on sophisticated admission control and resource reservation mechanisms in order to provide guarantees or statistical assurances for absolute performance measures, such as minimum service rate or maximum end-to-end delay. Recently there have been some new proposals for QoS in the Internet [3][4] that are based on a different model: in these architectures relative performance levels instead of absolute ones are provided (e.g. a high priority user can be guaranteed twice more bandwidth than a low priority one). Due to the nature of these schemes, admission control can be omitted, since the desired relative differentiation can always be achieved independent of the incoming load.

The absolute performance model is well suited for real-time traffic, which requires a specific capacity. However, we argue that the requirements of elastic traffic in a local network are better met by the relative performance model. Elastic applications do not require a specific capacity; instead, they use as much capacity as possible, and can still work at low data rates. Therefore, requesting a specific capacity does not match the nature of such applications.

This contribution addresses the issue of QoS support in W-LAN by extending the MAC protocol of the IEEE 802.11 standard. The proposed extension is divided in two steps. In the first step, real-time traffic is distinguished from elastic traffic by a priority scheduling approach and serviced according to the absolute performance model. In the second step, a service differentiation based on the relative performance model is introduced for elastic traffic.

The rest of the paper is structured as follows. In Section II we present the general ideas of our architecture. The details of the algorithms used in our architecture (for real-time and elastic traffic) are thoroughly described in Sections III and IV, respectively. In Section V we present our simulation results. The paper closes with the conclusions section.

II. Architecture

This section describes the extensions we propose for the MAC protocol of the 802.11 standard. Note that the scheduling resulting is equivalent to the scheduling of [3] in wired networks, with the difference that while [3] works with centralized queues, our protocol has to work in a distributed basis.

0-7803-6711-1/01/$10.00 (C) 2001 IEEE

A. Real-time extension

In contrast to elastic traffic, real-time packets are very sensitive to delays. In order to minimize the delay experienced by these packets, they should be given a prioritized access over the other packets.

The only solution in the current 802.11 MAC protocol that allows such a handling is the PCF mode, which receives a prioritized access to the medium over the DCF mode by using a shorter IFS. In our proposed extension for real-time traffic, we redefine the PCF function of the current standard into a distributed scheme. We argue that distributed control for supporting real-time services is more efficient and flexible than centralized control. The original PCF is not widely supported in current products, and the only requirement of our solution is that the original PCF must not be used in a network together with the extension presented here.

Redefining the PCF mode for real-time allows stations with real-time traffic to access the channel for the transmission of their packets after the PIFS, while stations with elastic packets have to wait until the end of the DIFS. In this way, real-time traffic receives a prioritized access over elastic traffic: whenever there is a real-time packet to be transmitted, it is always transmitted before any other packet.

The mechanism explained so far solves the contention between real-time and elastic packets by giving a higher priority to the former. However, different stations with real-time traffic may still collide when trying to access the channel after the PIFS. For this reason, a contention resolution algorithm is needed in order to avoid collisions between stations with real-time traffic. This algorithm is explained in detail in Section III.

Admission control is a key aspect for the real-time mechanism to work well, since it allows to limit the amount of real-time traffic admitted to the Wireless LAN. If there is too much real-time traffic, the resolution of contention in the redefined PCF will take too long and the requirement for immediate delivery of real-time packets will not be met. In addition, admission control can also be used to avoid starving elastic traffic types.

The effect of the admitted amount of real-time traffic to the delay experienced by real-time packets and to the capacity left for elastic traffic has been quantified via simulation (see simulation results in Section V). An exact design of an admission control scheme for our wireless architecture, however, still requires further investigation.

B. Elastic traffic extension

Elastic applications are tolerant of delays and experience an enhancement in performance as throughput is increased. Therefore, service differentiation for elastic traffic can be achieved by assigning a higher throughput to the users with a higher priority.

The elastic traffic extension we propose is based on the notion of a *share* that each user gets assigned. The *share* of a user reflects the level of QoS that the user gets, in such a way that the throughput experienced by a user i, r_i, is proportional to the *share* that this user has been assigned, s_i:

$$\frac{r_j}{s_j} = \frac{r_i}{s_i} \quad \forall i, \forall j \tag{1}$$

The *share* is configured via administration. The basic service corresponds to *share* equal to 1, and any higher values mean "better than average" kind of treatment.

Another way of looking at the concept of *share* is to say that the total bandwidth available in the network is divided by the sum of the shares of all the users and then each one gets the amount of bandwidth corresponding to its *share*. So if for example the total throughput is 1 Mbit/s and there is one station with share 2 and eight stations with share 1, the first one gets 200 Kbit/s and the rest 100 Kbit/s each.

In the DCF approach, the throughput received by a station depends on its CW: the smaller the CW, the higher the throughput. In our proposal, we use a modified method for the calculation of the size of the CW in order to offer different service levels in terms of throughput. Having different CW gives on average more throughput to the higher priority stations, but provides no guarantee for each specific packet[1]. In Section IV we present in detail the algorithm for computing the CW for the elastic traffic extension.

Thus, elastic traffic uses the DIFS for accessing the channel and requires minor changes in the standard. This is desirable, since it is a simpler solution than designing a completely new protocol and, in addition, it ensures backward compatibility.

Note that the proposed extension for elastic traffic does not require admission control, in contrast to the extension for real-time traffic described before.

C. Backward Compatibility

According to the above explanation of elastic traffic, terminals conforming to the IEEE 802.11 standard and elastic traffic compete with each other with different CW. In order to allow backward compatibility, the stations conforming to the IEEE 802.11 standard should behave as elastic traffic stations with the default *share* (i.e. a *share* equal to 1). Therefore, the value of the CW for elastic traffic stations with a *share* equal to 1 will have to be equal to the CW values specified in the IEEE 802.11 standard.

[1] Note that this fits well the differentiation needs of elastic traffic, but it would not fit in the delay guarantee required by real-time traffic.

D. Protocol Operation

The combination of the mechanisms for real-time and elastic traffic explained above lead to the protocol operation shown in the example of Figure 1.

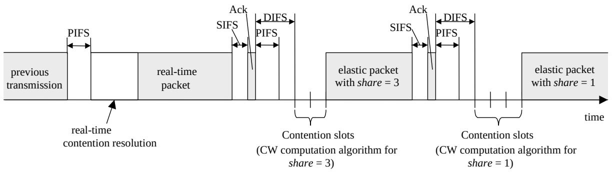
Ack
SIFS
DIFS
PIFS
PIFS
Ack
SIFS
DIFS
PIFS
previous transmission
real-time packet
elastic packet with *share* = 3
elastic packet with *share* = 1
time
real-time contention resolution
Contention slots (CW computation algorithm for *share* = 3)
Contention slots (CW computation algorithm for *share* = 1)

Fig. 1. Protocol Operation.

In this example, after the end of previous transmission, one station has a real-time packet to transmit. It accesses the channel, at the end of the PIFS. In order to make sure that collisions with other stations accessing the channel for real-time traffic are resolved, an additional contention resolution scheme is applied. After the end of the transmission, the receiver answers with an acknowledgement after a SIFS.

In the next access cycle, there is no real-time traffic to be transmitted, so the channel can be accessed by elastic traffic. In the example, it is an elastic packet of a user with a *share* equal to 3 that accesses the channel. The packet waits for the end of the DIFS and another two contention slots before it starts its transmission. As commented before, this elastic packet fully complies with the existing DCF MAC scheme, but has smaller CW, according to the *share* of its user. The receiver again answers with an ACK. Finally, an elastic packet of a user with a *share* equal to 1 is transmitted.

III. Contention Resolution Algorithm for the Real-time extension

The principle of the contention resolution scheme for the real-time extension is shown in Figure 2. The scheme is designed such that the number of collisions is minimised. A similar scheme is described in [5]. According to [5], the residual collision rate of the scheme is around 3.5% and is almost independent from the number of contending stations.

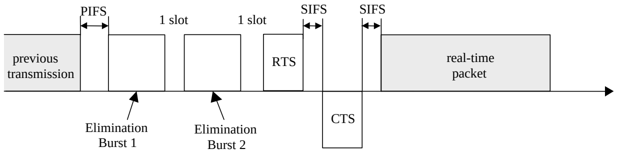

Fig. 2. Contention resolution scheme for the real-time extension.

A station with real-time traffic starts its contention cycle after a PIFS has passed after the end of a previous transmission. It uses two bursts for elimination, elimination burst (EB) 1 and EB 2. The duration of the EBs is a multiple of the Slot Duration defined in the 802.11 standard. The duration of EB 1 is calculated according to the following probability density:

$$P_{E1}(n) = p_{E1}^{n-1}(1-p_{E1}) \quad \text{for } 1 \leq n < m_{E1} \qquad (2)$$

$$P_{E1}(n) = p_{E1}^{m_{E1}-1} \quad \text{for } n = m_{E1}, \qquad (3)$$

where n is the number of slot durations EB 1 shall last, p_{E1} is a probability parameter between 0 and 1 and m_{E1} is the maximum number of slots EB1 can last. Note that the above formula requires that EB1 lasts at least one slot. This is necessary in order to occupy the channel and keep terminals with elastic traffic from making an access attempt.

The duration of EB 2 shall be calculated according to the probability density

$$P_{E2}(n) = \frac{1}{m_{E2}} \quad \text{for } 1 \leq n \ m_{E2}, \qquad (4)$$

i.e. it is taken from an equally distributed variable in the range between 1 and the maximum number of EB 2 slots, m_{E2}. Note that here the duration is at least one slot for the same reasons as for EB 1.

A station that makes an access attempt first chooses the duration of EB 1 and EB 2. If it senses the channel free for at least a PIFS, it transmits its EB 1 in any case. After this transmission, the station senses the channel. If it is free, it continues to send its EB2 after a slot duration. After the transmission of EB 2, it senses the channel again. If it is free, it starts to transmit its RTS or data packet after a slot duration and the transmission continues as defined for the data transmission using the DCF. If, however, the stations senses the channel busy after its transmission of EB 1 or EB 2, it withdraws its transmission attempt and defers until the channel has been free for at least a PIFS. Using this mechanism, the station which chooses the longest EB 1 or EB 2 among all contending stations wins the competition and is allowed to transmit.

If two stations happen to have the same EB 1 and EB 2 durations, they collide. However, due to the importance of the packets, we use the already defined mechanisms in 802.11 for collision detection, i.e. either RTS/CTS or the transmission of an ACK after the reception. In fact, the ACK will be transmitted in any case if a packet is being transmitted from a station using the new scheme to a terminal using the old scheme and, hence, shall be kept for the sake of backwards compatibility.

A similar scheme is investigated in [5]. The analysis there shows that the scheme is extremely stable and the residual collision rate is very low even for very high numbers of stations. The residual collision rate is almost independent from the number of contending stations. It depends, however, from the parameters p_{E1}, m_{E1} and m_{E2} given above. Another remarkable property of this scheme is that delays are very low up to a certain offered load but rise rapidly beyond this value.

IV. Contention Window Computation for the elastic traffic extension

In the DCF mode, the size of the CW determines the probability for a station to win the contention. The smaller the CW is, the higher the probability of getting access to the channel. As a consequence, there is a direct relationship between the CW assigned to a station and the bandwidth that this station will receive in a specific scenario. The condition expressed in Equation 1 for elastic traffic differentiation can thus be fulfilled by assigning to a station a CW value according to *share* of its user. The difficulty of this approach, however, relies in determining the CW values that will lead to the specified relative throughputs.

The approach we have chosen for the calculation of the CW for the elastic traffic extension is a dynamic one. In order to be able to properly adjust the CW (for each station to get its share), we introduce a variable w_i defined as

$$w_i = \frac{r_i}{s_i} \qquad (5)$$

where r_i is the estimated throughput experienced by the station and s_i is the *share* assigned to its user. The estimated throughput, r_i, is updated every time a new packet is transmitted:

$$r_i^{new} = (1 - e^{-\Delta t_i/k})\frac{l_i}{\Delta t_i} + e^{-\Delta t_i/k} r_i^{old} \qquad (6)$$

where l_i and Δt_i are the length and interarrival time of the transmitted packet, and k is a constant (in our case k is equal to 0.15^{-1}).

With the above definition of w_i, the resource distribution expressed in Equation 1 can be achieved by imposing the condition that the variable w_i should have the same value for all the stations:

$$w_i = w \quad \forall i \qquad (7)$$

Note that the actual value of w can vary in time (depending on the number of users for example). Note also that it is not necessary to know the actual throughput of the network: regardless the speed of the used Wireless LAN, the algorithm remains the same and works without any modification.

Equation 7 is fulfilled by using the following algorithm: having calculated its own w_i, each station includes it in the header of the packets it sends. For each observed packet, if the w_i in the packet's header is smaller than the w_i of the station, the station increases its CW by a small amount, while in the opposite case the station decreases its CW by a small amount. In this way, the w_i of all the stations tend towards a common value, w.

The above explanation describes the basics of the algorithm. However, in the adjustment of the CW, there are additional aspects that have to be taken into account:

- We do not want the CW to increase above the values defined by the 802.11 standard; as argued in Section II, for the backward compatibility reasons the basic service (with a *share* equal to 1) uses the CWs defined in the 802.11 standard, and any higher *share* should receive a "better than average" kind of treatment and therefore the values of the CW should be lower or at least equal.
- If the low sending rate of the application is the reason for transmitting below the desired rate, then the CW should obviously not be decreased. This can be detected by the fact that in this situation the transmission queue is empty.
- CWs should not be allowed to decrease in such a way that they negatively influence the overall performance of the network. If the channel is detected to be below its optimum limit of throughput due to too small values for the CWs (i.e. overutilization), the CW should be increased.

The above considerations lead to the following algorithm for the computation of the CW for each observed packet:

```
if ($w_{own} > w_{rcv}$) then $CW = (1 + \Delta_1)CW$
else if ($queue\_empty$) then $CW = (1 + \Delta_1)CW$
else $CW = (1 - \Delta_1)CW$
$CW_{Min802.11} \leq CW \leq CW_{Max802.11}$        (8)
```

where w_{own} is the value of w_i calculated by the station, w_{rcv} is the value of the w_i field in the observed packet and Δ_1 is computed as follows:

$$\Delta_1 = k\left|\frac{w_{own} - wrcv}{w_{own} + wrcv}\right| \qquad (9)$$

where k is a constant equal to 0.01.

So far we have not discussed one important issue which is the overutilization. In fact, due to the nature of our protocol and in particular due to the dynamic way of adjustment of the size of the CW, a mechanism for controlling the overutilization is necessary. This mechanism has been described in [7]

V. Simulations

To test the performance of the architecture presented in this paper, we have simulated it on a network consisting of a number of wireless terminals in a 2 Mbit/s Wireless LAN. These simulations have been performed in ns-2 [6].

Table I shows which combinations of numbers of real-time stations and data rate per real-time station meets

	data rate (kbps)				
stations	32	64	128	256	512
2	x	x	x	x	x
4	x	x	x	x	
6	x	x	x		
8	x				
10					

TABLE I
OVERVIEW WHICH CONFIGURATIONS MEET THE QUALITY CRITERION

the quality criterion. Considering the quality criterion a maximum delay of 25 ms for real-time traffic and that this limit should not be exceeded by at least 97 % of the packets. A cross in the table entry means that the criterion is met. As a rule of thumb, an admission control derived from this table would allow not more than six stations and not more than a data rate of 128 Kbit/s per station for real-time traffic. Delay distribution properties have been studied for a varying numer of real-time stations and for different source rates, detailed results are provided in [7].

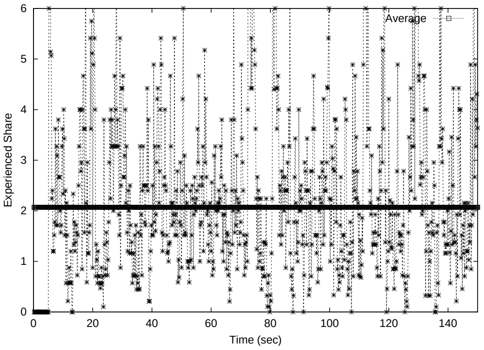

Fig. 3. Instantaneous Bandwidth for Elastic traffic.

The relative differentiation for elastic traffic, in our proposal, is done adjusting adaptively the CW of elastic traffic stations according to the measured performance. Figure 3 shows this dynamic adjustment; the simulations correspond to a scenario with a total number of 10 stations, 2 of them with a *share* of 2 and the rest with a *share* of 1. All stations are sending UDP CBR traffic with a packet size of 1000 bytes. It can be seen when comparing the instantaneous bandwidth of high priority and a low priority station that their ratio oscillates around the desired value (note that the average *experienced share* depicted in the figure is almost equal to 2, which is the *desired share*). A detailed simulation study of the elastic traffic as a function of the *shares* and the total number of stations for constant bit rate, bursty and TCP sources can be consulted in [7].

VI. Conclusions

In this paper we have presented a novel architecture for QoS support in wireless LAN. The proposed architecture is based on the IEEE 802.11 standard, and has been designed with the goal of minimizing the migration effort from this standard.

The architecture presented consists of two extensions: one for real-time traffic and another for elastic traffic. These extensions are adapted to the different natures of the two traffic types, providing absolute performance levels for real-time traffic and relative performance levels for elastic traffic. We argue that the requirements of the applications are better satisfied with these models.

The real-time extension redefines the PCF mechanism of the 802.11 standard into a distributed scheme. We argue that distributed control is more efficient and flexible than centralized control. Our distributed scheme does not provide hard QoS guarantees for individual packets. Note, however, that if we assume a soft QoS architecture like Diffserv in the backbone, it is actually not useful to provide harder QoS guarantees in the wireless access. Simulations have proved that with proper admission control the proposed extension for real-time traffic can provide very good statistical guarantees.

The extension for elastic traffic differentiation modifies the CW computation of the DCF mode of the standard. The simulations performed show that with this extension the desired level of differentiation is achieved in a wide variety of scenarios. The modification to the DCF mode has been done in such a way that the proposed architecture is backwards compatible, i.e. terminals conforming to the current standard are supported. The impact of 802.11 terminals to the proposed architecture has also been studied via simulation.

References

[1] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview," RFC 1633, June 1994.

[2] K. Nichols, V. Jacobson, and L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet," RFC 2638, July 1999.

[3] A. Banchs and R. Denda, "A Scalable Share Differentiation Architecture for Elastic and Real-time Traffic," Proceedings of 8th International Workshop on Quality of Service IWQoS'2000, June 2000.

[4] C. Dovrolis and P. Ramanathan, "A Case for Relative Differentiated Services and the Proportional Differentiation Model," *IEEE Network*, vol. 12, no. 5, pp. 26–34, September 1999.

[5] S. Chevrel, A. Aghvami, et al. "Analysis and optimisation of the HIPERLAN Channel Access Contention Scheme," Wireless Personal Communications 4, pp. 27-39, Kluwer Acadamic Publishers, 1997.

[6] "Network Simulator (ns), version 2," http://www-mash.cs.berkeley.edu/ns.

[7] "Service Differentiation Extensions for Elastic and Real-Time traffic in 802.11 Wireless LAN" http://www.icsi.berkeley.edu/~banchs/diffext80211.pdf.