Providing Throughput Guarantees in IEEE 802.11 Wireless LAN

Albert Banchs, Xavier Pérez
NEC Europe Ltd., Network Laboratories Heidelberg, Germany

Abstract— **In this paper, we propose ARME (Assured Rate MAC Extension), an extension of the IEEE 802.11 MAC protocol to provide throughput guarantees. The proposed extension relies on the Distributed Coordination Function (DCF) with a modified algorithm for the computation of the Contention Window (CW). Best Effort service (with no throughput guarantee) is supported by the functionality of the current 802.11 standard in such a way that legacy IEEE 802.11 terminals behave as Best Effort terminals in ARME. The performance of the proposed extension has been extensively evaluated through simulation; simulation results show that IEEE 802.11 devices using ARME behave well for different types of traffic and different source rates.**

Index Terms—**Wireless LAN, Throughput Guarantees, Assured Rate Service, Differentiated Services, Quality of Service, MAC, IEEE 802.11**

I. Introduction

One of the biggest challenges in today's computer networks is to provide the Quality of Service (QoS) appropriate for the constantly growing demand from the side of applications. Over the last ten years, considerable effort has been made to provide QoS to the Internet, with proposals such as Integrated Services [1] and Differentiated Services [2]. Both of these architectures use queuing mechanisms which schedule and drop packets according to their delay priority and bandwidth assurance.

QoS mechanisms are of particular relevance in the case of Wireless LAN, where the bandwidth is scarce and the efficient use of it is of special importance. Frequency is a scarce resource and, due to the propagation characteristics of the radio channel, is a shared medium for those using it.

Since Wireless LANs may be considered as just another technology in the communications path, it is desirable that the architecture for QoS support follows the same principles in the wireless network as in the wireline Internet, assuring compatibility among the wireless and the wireline parts. The Differentiated Services (DiffServ) architecture for the wireline Internet aims at providing simple and scalable service differentiation by discriminating and treating the data flows according to their service class [2]. DiffServ makes a trade-off: QoS for individual packets is not necessarily guaranteed, but the DiffServ architecture scales well and is easy to implement. Because of these reasons, DiffServ is an increasingly popular approach for providing QoS in the Internet.

DiffServ standardization is currently an ongoing effort. Up to date, two Per-Hop Behaviors (PHBs) have been standardized: the Expedited Forwarding PHB [3] and the Assured Forwarding PHB [4], and several Per-Domain Behaviors (PDBs) have been proposed for standardization: the Virtual Wire PDB [5], the Bulk Handling PDB [6] and the Assured Rate PDB [7], [8].

This paper proposes an Assured Rate Service Extension for the MAC layer of the IEEE 802.11 standard (ARME: Assured Rate MAC Extension), in line with the Assured Rate PDB proposed for DiffServ. This Assured Rate Service guarantees a specific throughput to its user. A typical user of this service could be the CEO of a company requiring a high speed access to Internet independent of the level of congestion of the company's Wireless LAN.

The rest of the paper is organized as follows. In Section II we recall the basics of the IEEE 802.11 standard. In Section III we explain the ARME architecture for an Assured Rate Service and its interaction with the Best Effort Service and legacy 802.11 terminals. The algorithm used in ARME for the Contention Window (CW) computation is thoroughly described in IV. In Section V we present our simulations results and, finally, the paper closes with an overview on related work and the conclusions (Sections VI and VII).

II. The IEEE 802.11 MAC layer

The basic IEEE 802.11 Medium Access mechanism is called Distributed Coordination Function (DCF) and is based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol [9]. CSMA/CA was first investigated thoroughly in [10] and [11]. The MAC scheme used in IEEE 802.11 is an extended version of the FAMA protocol [12]. It is slotted, i.e. the access can happen only at specific instants. The 802.11 MAC protocol operation is shown in Figure 1.

In the DCF mode, a station must sense the medium before initiating the transmission of a packet. If the medium is sensed idle for a time interval greater than the DCF Inter Frame Space (DIFS), then the station transmits the packet. Otherwise, the transmission is deferred and a backoff process is started.

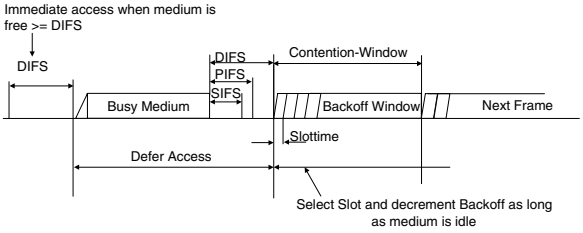

Fig. 1. Basic 802.11 MAC protocol operation

Specifically, the station computes the backoff interval as an equally distributed random value taken from the range of 0 to the so-called Contention Window (CW), where the backoff time is measured in slot times. This backoff interval is then used to initialize the backoff timer. This timer is decreased only when the medium is idle and is frozen when it is sensed busy. Each time the medium becomes idle for a period longer than a DIFS, the backoff timer is periodically decremented, once every slot-time.

As soon as the backoff timer expires, the station starts to transmit. A collision occurs when two or more stations start transmission simultaneously in the same slot. To avoid collisions, a Request To Send (RTS) and a clear to send (CTS) can be exchanged between source and receiving station prior to the actual frame transmission. In addition, an Acknowledgement (Ack) is transmitted to the source after successful reception of the frame to detect collisions. The Ack scheme can additionally be used to control the retransmission of erroneous frames. The RTS/CTS scheme is also used for hidden node handling.

If a CTS or acknowledgment is not received by the source station, it assumes that the transmission attempt was not successful and re-enters the backoff process. To reduce the probability of collisions, the CW is doubled after each unsuccessful transmission attempt until a predefined maximum (CW_{max}) is reached. After a successful frame transmission, if the station still has frames buffered for transmission, it must execute a new backoff process.

The second access mechanism specified in the IEEE standard is built on top of DCF and it is called Point Coordination Function (PCF). It is a centralized mechanism, where one central coordinator polls stations and allows them undisturbed, contention free access to the channel. In contention free access mechanism, collisions do not occur since the access to the channel is controlled by one entity. The PCF mechanism, however, is not supported in most wireless cards, and it was shown in [13] that the cooperation between PCF and DCF modes leads to poor throughput performance. The PCF scheme has no practical meaning to this paper.

The three Inter Frame Spaces (IFS) serve the purpose of defining different levels of access priorities. They define the minimal time that a station has to let pass after the end of a frame, before it may start transmitting a certain type of frame itself. After a SIFS (Short IFS), the shortest interframe space, only acknowledgements, CTS and data frames in response to poll by the PCF may be sent. The use of the PIFS and the DIFS serves to separate the PCF and DCF modes, giving a higher priority to the former.

III. Assured Rate MAC Extension (ARME)

DiffServ is based on simple mechanisms with minimal control and signaling, and does not require to keep per-flow state at core nodes. ARME, the Assured Rate MAC Extension we propose, follows the same principles: it is based on distributed control, minimizing thus the signaling overhead at the MAC layer, and does not require to keep per-flow state at the MAC level. Note that the current 802.11 MAC is also distributed and connectionless. The introduction of a centralized and connection-oriented MAC scheme as an extension to 802.11 would be a major change in the paradigm and would probably impact the backward compatibility and increase the migration effort.

Also like DiffServ, the ARME architecture provides a soft kind of QoS, i.e. statistical QoS guarantees are given to traffic aggregates, but an individual packet does not receive any kind of guarantee. Note that this fits well the type of QoS that can be achieved with a distributed and connectionless MAC.

In ARME we distinguish two types of service: the Assured Rate Service and Best Effort. An Assured Rate station in ARME is a station that has contracted a service with a certain assured rate, while a Best Effort station has not contracted any rate. In the discussion and simulations of this paper, we assume that each station is using only one service (Assured Rate or Best Effort). The proposed algorithm, however, can be easily extended when one node is using both services.

In the DCF approach, the throughput received by a station depends on its CW: the smaller the CW, the higher the throughput. In ARME, the Assured Rate Service is supported by the DCF function of the current standard with minor changes in the computation of the CW in order to give to each station the expected throughput according to the service contracted by the station. Thus, both the Assured Rate station and the Best Effort access the channel with the DCF mode but with different CWs. In Section IV we present in detail the algorithm for computing the CW for Assured Rate stations.

The protocol operation of ARME is shown in the example of Figure 2, in which, after the end of a previous transmission, there are two stations with a packet to transmit, one Assured Rate station and one Best Effort station. In the example, the Assured Rate station, which is competing with a smaller CW, accesses the channel first. The Best Effort station uses the CW calculated according to the current IEEE 802.11 standard and accesses the channel afterwards. Note that with this choice of the CW for Best Effort, 802.11 terminals behave as Best Effort terminals in the ARME architecture, providing thus backward compatibility.

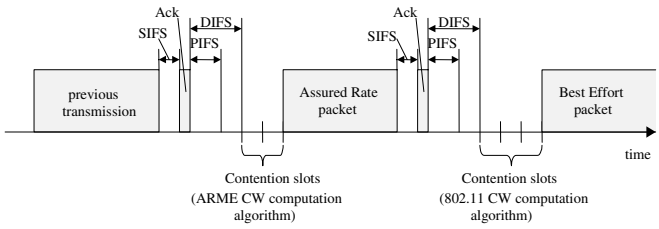

Fig. 2. Protocol Operation.

According to the above explanation of ARME, Best Effort and Assured Rate packets use the same IFS but compete with each other with different CWs. This can be compared to the RIO buffer management in DiffServ [14], in which *in-profile* and *out-of-profile* packets share the same buffer but are dropped with different probabilities as a function of the buffer occupancy. The difference is that ARME, in contrast to RIO, has to work on a distributed basis.

The CW of Best Effort traffic cannot be arbitrarily increased for backward compatibility reasons. Also, the CW of Assured Rate traffic cannot be arbitrarily decreased, since this would lead to an unstable situation with permanent collisions. The consequence of these limitations in the CWs make it impossible to totally control the capacity given to each service. Therefore a certain level of impact of Best Effort to Assured Rate is unavoidable. This impact has been studied in the simulations (see Section V-B).

Our approach requires admission control to ensure that the sum of the throughputs committed to the Assured Rate Service is not larger than the total throughput available in the Wireless

LAN. This admission control in the wireless access should be considered as an integral part of the admission control defined in the DiffServ architecture.

IV. Contention Window Computation for ARME

In the DCF mode of the 802.11 standard, 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. An Assured Rate Service can therefore be provided by assigning to a station the CW corresponding to the bandwidth requested by this station.

The difficulty of this approach, however, relies in determining the CW that will lead to the specified bandwidth. Note that this value depends on the number of stations that compete for accessing the channel and their CWs, which is a changing condition.

A. Contention Window Computation

The approach we have chosen for the calculation of the CW in ARME is a dynamic one: each station monitors the bandwidth experienced and modifies its CW in order to achieve the desired throughput. For each packet transmitted, we estimate the sending rate of the terminal; in the case that the estimated rate is smaller than the desired one, we slightly decrease the CW, while in the opposite case, we increase it slightly.

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 used by the Best Effort terminals, since this would lead to a worse performance than Best Effort. On the other hand, as explained in Section III, for backward compatibility reasons, the CW for Best Effort should be the one defined by the 802.11 standard.
- 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.
- When estimating the sending rate, it would be desirable to control the allowed burstiness of the source.
- CWs should not be allowed to decrease in such a way that they negatively influence the overall performance of the network.

Considering all the above issues, we have designed an algorithm for the computation of the CW, which is inspired in the token bucket algorithm. In our scheme, we use the number of bytes in the bucket (bucket length) and the occupancy of the transmission buffer (queue length) as input parameters in the algorithm (see Figure 3). This is further explained in the following points:

- The token bucket gets filled at the desired transmission rate. For each successful transmission, the length of the transmitted packet in bytes is subtracted from the bucket. Thus the bucket length ($blen$) represents the resources that the user has for transmitting packets.

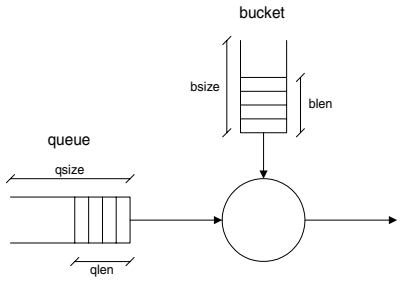

Fig. 3. Token bucket algorithm for AS.

- The user has resources to transmit a packet only if the bucket has enough bytes in it (we have taken a certain limit $blim$ to represent the minimum needed).
- The bucket size ($bsize$) determines the accepted burstiness of the source; the maximum length allowed to a burst is equal to $bsize - blim$.
- The queue length ($qlen$) expresses the willingness of a station to transmit packets. The CW is only decreased if the queue is not empty (if the queue is empty, the user is not filling it, which means that the current CW satisfies the sending needs of the user).
- When increasing the CW, the value assigned to it can never exceed the size of the CW used for Best Effort.
- If the channel is detected to be below its optimum limit of throughput due to too small values for the CWs (i.e. overload), the CW is increased. This aspect is discussed in detail in the following clause.

The above considerations lead to the following algorithm. This algorithm computes a value p which is used to scale the CW values defined in 802.11. Note that, besides this scaling of the CW, the backoff time computation algorithm is left as defined in the 802.11 standard (i.e. the Contention Window is doubled after each unsuccessful transmission attempt for a given number of times).

$$\begin{gathered}
\texttt{if } (qlen = 0) \texttt{ then } p = (1 + \Delta_1)p \\
\texttt{else if } (blen < blim) \texttt{ then } p = (1 + \Delta_2)p \\
\texttt{else } p = (1 - \Delta_3)p \\
p = min\{p, 1\} \\
CW = p \cdot CW_{802.11}
\end{gathered} \tag{1}$$

where Δ_1 is a constant and Δ_2 and Δ_3 are calculated in the following way

$$\Delta_2 = \frac{blim - blen}{blim}\Delta_1 \tag{2}$$

$$\Delta_3 = \frac{blen - blim}{bsize - blim}\Delta_1 \tag{3}$$

The presented algorithm depends on the number of parameters, namely $blim$, $bsize$ and Δ_1. Simulations have shown that the tuning of these constants is not critical for the performance of the protocol as long as they have reasonable values. In the simulations results presented in Section V we have taken $blim$ equal to $token\_size$, $bsize$ equal to $5 * token\_size$, $token\_size$ equal to 1072 bytes and Δ_1 equal to 0.025.

B. Overload

So far we have not discussed one important issue which is the overload. In fact, due to the nature of our algorithm and, in particular, due to the dynamic way of adjustment of the size of the CW, a mechanism for controlling the overload is necessary.

As we can see in (1), each station adjusts its CW only on the basis of its own requirements. Such "selfishness" can lead to an unstable state, due to the following side effect of the CWs. We have been arguing so far that, the smaller the CW for a given station, the bigger the probability for this station of seizing the channel before any other station. But another consequence of such a procedure is that the more stations with a small CW, the bigger the probability of a collision. If there is a large number of Assured Rate stations, this can lead to an absolute blockage of the channel. Once all of the stations start decreasing their CWs in order to get the requested bandwidth, the number of collisions will start increasing, and this will decrease the overall throughput of the channel, and, as a consequence, the bandwidth experienced by each station. This will lead to even smaller CWs, and therefore, to an unstable state with continuous collisions. A solution to avoid this situation, which we have called *overload*, is to extend (1) with the following condition:

$$\begin{gathered}\texttt{if } (overload) \texttt{ then } p = (1+\Delta_4)p \\ \texttt{else if } (qlen = 0) \texttt{ then } p = (1+\Delta_1)p \\ \texttt{else if } (bsize < blim) \texttt{ then } p = (1+\Delta_2)p \\ \texttt{else } p = (1-\Delta_3)p \\ p = min\{p, 1\} \\ CW = p \cdot CW_{802.11} \end{gathered} \quad (4)$$

where $\Delta_4 = 0.25$ is again a constant.

The above equation requires of some way to detect when we are in a situation of overload. As mentioned before, in a situation of overload each station experiences a large number of collisions. Therefore, if we now provide each station with a collision counter[1], which determines how many collisions in average a packet experiences before it is successfully transmitted, we can write the following simple condition to determine overload

$$\texttt{if } (av\_nr\_coll > c) \texttt{ then } overload = true, \quad (5)$$

where c is a constant that has to be properly adjusted. If c is too low, AS stations will not be allowed to decrease their CWs sufficiently, and as a consequence they will not be able to achieve the desired bandwidth. On the other hand, if c is too large, the number of collisions in the channel will be very high and the overall performance will be harmed. This constant, therefore, represents a tradeoff between the level of differentiation of AS against Best Effort and the efficiency (i.e. total throughput) of the channel. This tradeoff has been studied via simulation (see Section V-E), and an optimum value for c has been chosen according to simulation results.

[1]Note that in 802.11 collisions can only be detected through the lack of the Ack. However, a missing Ack can also be caused by other reasons different than a collision. In the simulations section we study the impact into our algorithm of having missing Acks due to errors in the channel (see Section V-F).

The average number of collisions, (av_nr_coll), in Equation 5 is calculated after each successful transmission in the following way

$$av\_nr\_coll = (1-t) * num\_coll + t * av\_nr\_coll \quad (6)$$

where in order to smoothen its behavior, we use some sort of memory, taking into account the last calculated value of av_nr_coll (on the rhs of Equation 6). The constant t is a small number (in our case $t = 0.25$) playing the role of a smoothening factor.

V. Simulations

To test the performance of the ARME scheme presented in this paper, we simulated it on a network consisting of a number of wireless terminals in a 2 Mbps Wireless LAN communicating with a fixed node. These simulations were performed in ns-2 [15]. For this purpose, the CW computation algorithm of Equation 4 was inserted into the existing implementation of the 802.11 MAC DCF protocol in ns-2. In the simulations performed, stations using the normal 802.11 MAC protocol (i.e. Best Effort in our architecture) coexisted with stations using the Assured Rate Service, in such a way that each station used either the Assured Rate Service or Best Effort. The packet length was set to 1000 bytes for all simulations.

We chose to use the RTS/CTS mechanism in all cases. This mechanism, optional in the 802.11 standard, increases bandwidth efficiency in case of many collisions, since with this mechanism collisions occur with the relative small control packets rather than with long data packets. Since our architecture may lead to larger number of collisions than the normal 802.11 MAC DCF, this mechanism can be especially beneficial in our case.

A. Bandwidth Assurance

In ARME, the assurance of the requested bandwidth for Assured Rate is done adjusting adaptively the CW of Assured Rate stations according to the measured throughput. Figure 4 shows this dynamic adjustment; the simulation corresponds to a scenario with a total number of 10 stations, 8 of which are Best Effort and 2 Assured Rate with a rate assurance of 500 Kbps each. All stations are sending UDP CBR traffic at a rate of 500 Kbps. It can be seen that the instantaneous bandwidth of Assured Rate stations (referred as AS in the graph) oscillates around the desired value (500 Kbps), while Best Effort stations (referred as BE) receive a much lower throughput.

B. Impact of Best Effort terminals

In Section III we have argued that it is impossible to avoid a certain level of impact of Best Effort stations on the Assured Rate Service. This impact is studied in the simulation results shown in Figure 5. This figure shows the variation of the throughput received by Assured Rate stations in different scenarios when the number of Best Effort stations increases. In these simulations, Assured Rate stations receive a bandwidth assurance such that a total amount of 1 Mbps is assigned to Assured Rate (i.e. in the case of 1 Assured Rate station, this

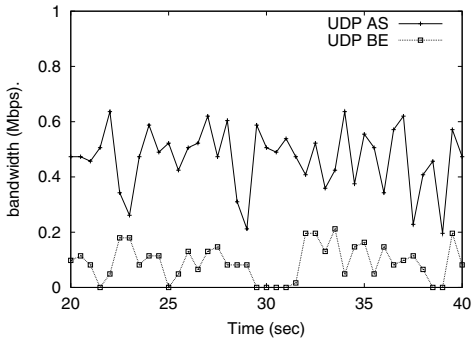

Fig. 4. Instantaneous Bandwidth of 1 AS station vs. 1 Best Effort station.

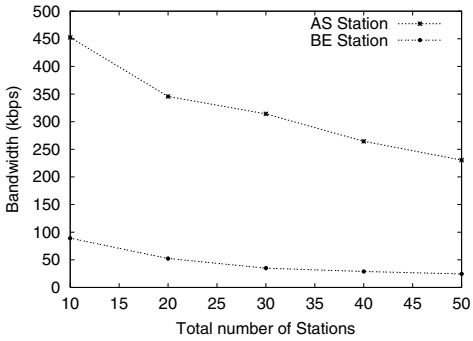

Fig. 6. Assured Service vs. Best Effort.

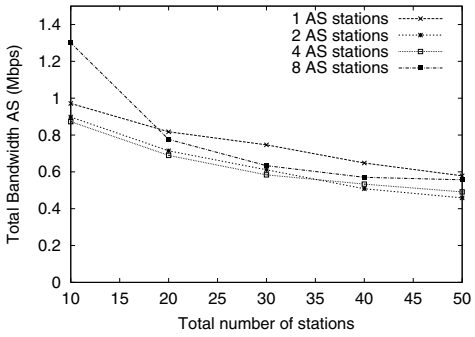

Fig. 5. Impact of Best Effort to the bandwidth for Assured Service.

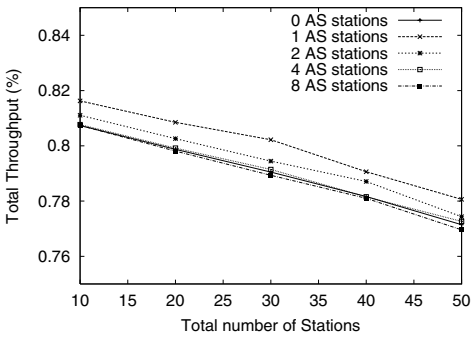

Fig. 7. Channel utilization.

station receives a bandwidth assurance of 1 Mbps; in the case of 2, each receives a bandwidth assurance of 500 Kbps; in the case of 4, 250 Kbps; and in the case of 8, 125 Kbps).

It can be seen that the total bandwidth received by the Assured Rate Service (ideally 1 Mbps shared among the Assured Rate stations) decreases with the number of Best Effort stations. When the total number of stations is 50, the bandwidth received by Assured Rate stations is about half of the committed rate (i.e. 500 Kbps). Note that the total bandwidth received by Assured Rate decreases with the total number of stations almost independently of the number of Assured Rate stations.

In the point corresponding to 8 Assured Rate stations and 2 Best Effort, Assured Rate receives a throughput much higher than the one committed (1.3 Mbps). Note, however, that if only the committed 1 Mbps was given to Assured Rate, the 2 Best Effort stations would experience each a higher throughput than an Assured Rate station, since they would share the remaining 1 Mbps. The nature of the mechanism we have proposed in ARME for the CW computation ensures that this undesirable situation does not occur: with our algorithm, the leftover bandwidth is equally shared between Assured Rate and Best Effort stations such that a Best Effort station never receives a better treatment than an Assured Rate one.

Figure 6 shows the bandwidth received by an Assured Rate station and the one received by a Best Effort station in the case of 2 Assured Rate stations and varying the total number of stations. It can be seen that it is not only Assured Rate stations but also Best Effort which see their bandwidth decreased when the total number of stations increases. Note that even though with 50 stations Assured Rate stations get about half of the committed bandwidth (250 Kbps each), they still get a throughput 10 times higher than Best Effort stations, which get about 25 Kbps each. This result could be interpreted as a good tradeoff between differentiation (Assured Rate stations get a much higher throughput) and fairness (Best Effort stations do not starve).

C. Channel utilization

Having Assured Rate stations with a CW smaller than the CW defined in the current standard can impact the channel utilization. Figure 7 shows the channel utilization for the same scenario than the described for Figure 5, and compares it to the channel utilization with the current standard (i.e. 0 AS stations).

It can be seen that the channel utilization increases as compared to the current standard when the number of Assured Rate stations is low. The reason for this is that, with no Assured Rate stations, the network is underloaded (i.e. the utilization is below the maximum basically due to too long idle times). Having one Assured Rate station with a CW lower than the Best Effort one, the load increases, which leads to a higher utilization. However, the utilization decreases with the number of AS stations. This is because having more than one Assured Rate station with a low CW leads to a situation of overload (i.e. too many collisions). In clause IV-B we have proposed an algorithm to control this effect, which is studied via simulation in clause V-E. With this algorithm, the channel utilization of ARME does not significantly decrease below the channel utilization of 802.11 even for a high number of Assured Rate stations, as can be seen in Figure 7.

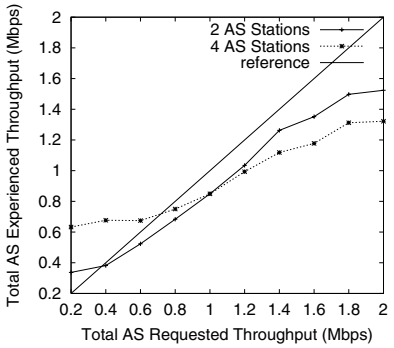

Fig. 8. Over and undercommitment.

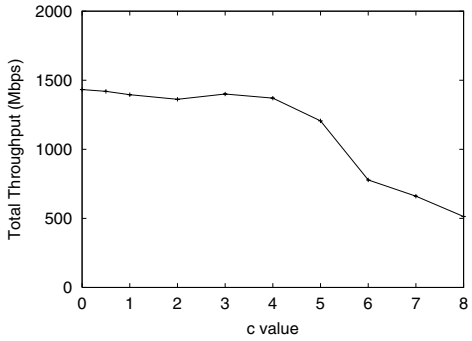

Fig. 9. Throughput as a function of c.

D. Over and undercommitment

The network is undercommitted when the Assured Rate Service is given a much lower bandwidth than the throughput available, and overcommitted when it is given a higher bandwidth than the available throughput.

A situation of undercommitment can occur when most of the Assured Rate stations are not active. In this case, it would be desirable that Assured Rate stations could use the leftover bandwidth, therefore receiving a higher throughput than the one requested.

On the other hand, a situation of overcommitment can easily occur when Assured Rate stations are configured statically. With static configuration it is desirable to be able to commit a higher bandwidth than the one available in the channel, assuming that it is highly improbable that all Assured Rate stations will be active at the same time. However, such configuration can lead to a situation of overcommitment with a certain probability. In such situation, it will be impossible for Assured Rate stations to receive the requested throughput. This situation, however, should not lead to unstability; instead, it would be desirable that Assured Rate stations shared the available overall data rate.

In Figure 8 it can be seen that the behavior with under and overcommitment is the desired. This simulation has been done for a scenario with a total number of 10 stations, of which 2 (first case) or 4 (second case) are Assured Rate stations and the rest are Best Effort stations. In the following clause a situation of extreme overcommitment has been simulated, showing that the behavior in that case is also the desired.

E. Impact of c

In clause IV-B, the constant c has been defined as the maximum average number of collisions allowed. This limit is needed in ARME to avoid loss of efficiency in case of overload due to too small CWs.

Since we are using the RTS/CTS mechanism, the number of collisions will never exceed 8 (according to the standard, a packet is dropped after 8 RTS tries). Therefore, the chosen value for c must be in the range of $0 < c < 8$.

The impact of c can be better analyzed in a scenario of extreme overcommitment, since overcommitment leads to a situation of overload. Therefore, to study the impact of c we have chosen to use a scenario consisting of a large number of stations (100 stations), half of them Assured Rate with a very high bandwidth assurance and the rest Best Effort. This scenario leads to many stations with very small CW, and, therefore a high number of collisions, that would block the channel were they not controlled by the parameter c. The reason for choosing this extreme scenario was to demonstrate that even in the limit of overload, the proposed algorithm avoids the blockage of the channel.

Figures 9, 10 and 11 show the total throughput, the number of drops per successful packet and the total throughput for Assured Rate and Best Effort as a function of c, for the scenario described above. In these figures it can be seen that if the value of c is too high, the total throughput experienced is very low, and the percentage of losses very high. In the extreme case ($c = 8$), the throughput reaches a very low value (500 Kbps) and the packet drops increase drastically. The reason for this is that, with such values of c, the CW can decrease too much and the collision probability gets too high. Note that in this case, Best Effort stations totally starve.

On the other hand, if the value of c is too low, we obtain a good total throughput and very low loss rate, but we do not achieve the desired differentiation between Assured Rate and Best Effort. In the limit ($c = 0$), there is no differentiation at all, and Assured Rate stations get exactly the same throughput as Best Effort. The reason for this is that, with such values of c, CWs are not allowed to decrease below the values for Best Effort (i.e. the ones defined in the 802.11 standard), and, therefore, the ARME extension defined in this paper is deactivated.

As a conclusion, c expresses a tradeoff between efficiency and differentiation, and it can be adjusted via network administration depending on specific user preferences. In this paper we have chosen to use an intermediate value: $c = 4$. With this value of c, a good level of differentiation is achieved, while conserving a good overall efficiency, even for the extreme scenario simulated in this clause.

F. Impact of Errors

The algorithm proposed in clause IV-B and simulated in the previous clause is based on a collision counter which counts every sent packet/RTS for which an Ack/CTS has not been received as a collision. However, in a non-ideal channel the lack of an Ack can also be due to an error in the channel; in this case, the lack of an Ack would be falsely interpreted by an Assured Rate station as an indication of overload. As a reaction,

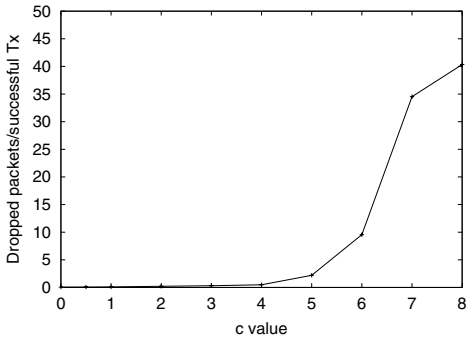

Fig. 10. Drops as a function of c.

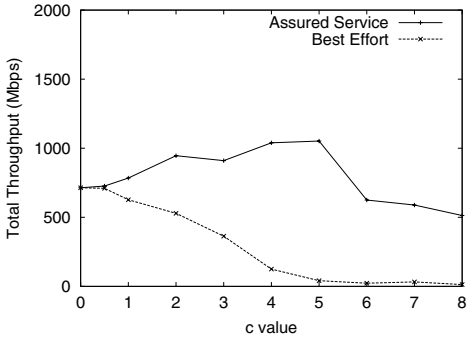

Fig. 11. Throughput for Assured Service and Best Effort as a function of c.

the AS station would increase its CW more aggressively than it should with the parameter c chosen, and, as a consequence, the achieved level of differentiation between AS and Best Effort would be lower than the desired.

Figure 12 shows the total throughput, the throughput of Assured Rate and the throughput of Best Effort as a function of the percentage of errors in the channel, for the same extreme scenario as in clause V-E with a value of c equal to 4. It can be seen that the level of differentiation (ratio between Assured Rate and Best Effort throughputs) decreases with the error rate, as expected. However, even at very high error rates (10%) in such a extreme scenario, the level of differentiation (i.e. the ratio between the throughput received by Assured Rate and Best Effort stations) still keeps reasonably high.

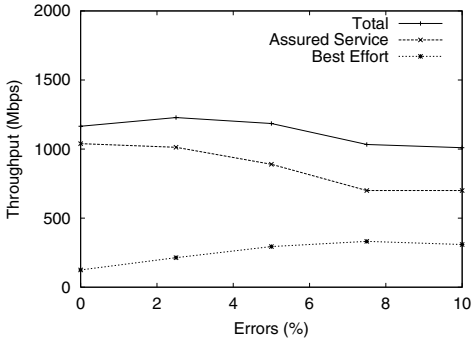

Fig. 12. Level of differentiation as a function of the error rate.

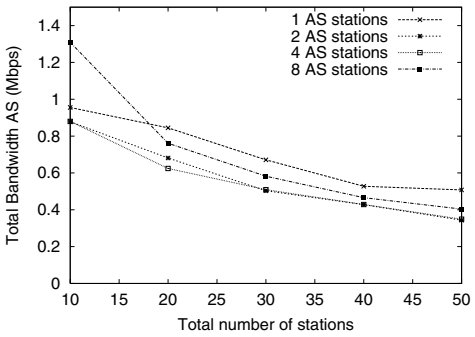

Fig. 13. Sources UDP ON/OFF 1 ms.

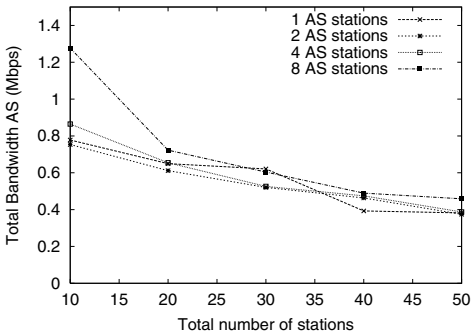

Fig. 14. Sources UDP ON/OFF 500 ms.

G. ON/OFF sources

The simulations shown so far correspond to a constant traffic (UDP CBR sources). In order to gain a better understanding of the impact of different traffic sources to the performance of ARME, we have simulated it under bursty traffic (UDP ON/OFF sources). Since we use a token bucket algorithm for the computation of the CW, the throughput received by a station will depend on the size of its burst. If the burst length is smaller than the bucket size, a burst does not empty the bucket and therefore the station does not see its throughput decreased; in contrast, if the burst length is larger than the bucket size, it empties the bucket and this results in a reduction of the throughput received by the station.

In order to show the impact of the burst size, we performed two different simulations: one with a small burst (ON/OFF periods of 1 ms in average), and one with large bursts (ON/OFF periods of 500 ms in average). The simulation scenario was the same as the described by Figure 5.

Figure 13 shows the results when the ON/OFF periods are of 1 ms. Note that these results are very similar to the results of Figure 5 (CBR traffic), which means that short ON/OFF periods do not impact the performance of a station, as argued above. In Figure 14 it can be seen that large ON/OFF periods, in contrast to short ones, do impact the experienced throughput. However, this impact is not too high. An explanation for this rather low impact could be that the internal buffering of the station performs as a traffic shaper, smoothening the bursty traffic.

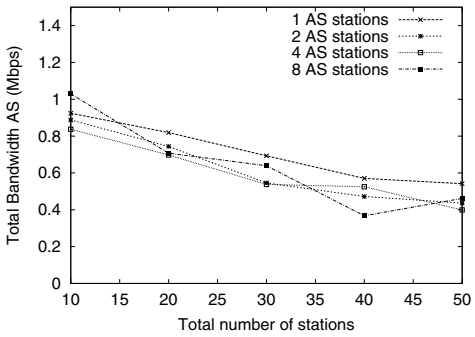

Fig. 15. Bandwidth for Assured Service with TCP Sources.

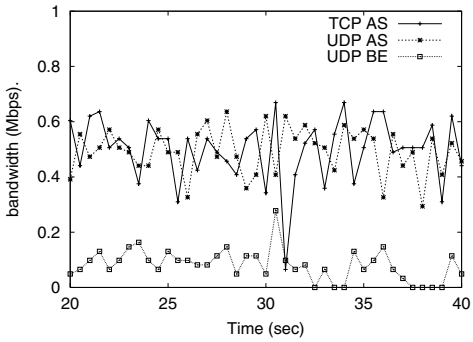

Fig. 16. TCP vs. UDP.

H. TCP sources

The Assured Rate Service aims at guaranteeing average bandwidth, which makes it especially suitable for data traffic. Since TCP is widely used for such traffic, the support of it is a key aspect of any architecture aiming at bandwidth assurance.

Figure 15 shows the bandwidth obtained by Assured Rate stations for the same experiment as in Figure 5, when the traffic sources are endless TCP. It can be seen that under TCP traffic the proposed architecture performs almost as well as with UDP CBR traffic. Our algorithm, therefore, can work together with the congestion control of a TCP source, providing TCP sources with the committed throughput.

Note that, in contrast to the previous experiments, in this case it is necessary to reserve bandwidth in the downlink for Assured Rate in order to achieve the desired bandwidth distribution, since the TCP acknowledgements in the return path determine through the congestion control of TCP the throughput of the flow. The main difference between the downlink and uplink channels is that in the downlink the queue is not distributed but centralized. The enqueuing algorithm used in order to achieve the desired behavior in the downlink channel is the one defined in the DiffServ architecture for wired networks [2].

The amount of bandwidth reserved for the downlink in the experiments of Figure 15 was of 300 Kbps. The determination of how much bandwidth has to be reserved in the return path is an issue of DiffServ and is beyond the scope of this paper.

I. TCP vs. UDP

When TCP and UDP flows compete with each other, the bandwidth distribution tends to favor UDP. This is because, in case of congestion, TCP backs off because of its congestion control mechanism, and UDP, without any kind of congestion control and therefore more aggressive, consumes the bandwidth left by TCP. A QoS architecture with bandwidth assurance should overcome this different level of aggressiveness of the sources and provide all sources with their committed throughput independent of the congestion control algorithm they use. This requirement, however, is difficult to meet, and most QoS architectures do not achieve the desired level of fairness between TCP and UDP (see e.g. [16] for the Differentiated Services architecture).

To study the level of fairness between TCP and UDP achieved by ARME, we performed the following experiment: two Assured Rate stations, one endless TCP and the other UDP CBR, had a committed throughput of 500 Kbps, while the remaining 8 stations were Best Effort stations sending UDP CBR traffic. The UDP Assured Rate station sends at a rate of 500 Kbps, the Best Effort ones at a rate of 1 Mbps. Figure 16 shows the instantaneous bandwidth achieved by the TCP Assured Rate source, the UDP Assured Rate source and one UDP Best Effort source. It can be seen that the result is the desired: both Assured Rate stations oscillate around their committed throughput, while Best Effort stations receive the bandwidth left over by Assured Rate.

From this experiment we conclude that ARME provides TCP with a fair treatment with respect to UDP. This is because the ARME algorithm adapts the CW to the aggressiveness of the source: a less aggressive source, like TCP, will see its CW reduced until it receives the committed throughput, while a more aggressive source, like UDP, will achieve its committed throughput with a larger CW.

VI. Related work

Current trends in wireless networks indicate a desire to provide a flexible wireless infrastructure that can support high quality services along with traditional best effort. In such a wireless environment, QoS support becomes critical.

One possible approach for supporting QoS in Wireless LAN is based on the Integrated Services architecture proposed for the wireline Internet [17]. In this approach, the control over wireless resources is very strict, motivated by the argument that strict control, with complex and sophisticated mechanisms and protocols, is required to maintain good quality in the wireless environment.

Another approach for QoS support in Wireless LAN is based on the Differentiated Services architecture, which provides service differentiation using more simple mechanisms. There have been several proposals for service differentiation in wireless networks, like in [18]. These mechanisms, however, rely on centralized control and polling of backlogged mobile hosts. In contrast to these proposals, the architecture we propose is based on distributed control. We argue that distributed control of radio resources results in a more productive use of radio resources.

[19], [20], [21], [22], [23] and [24] are other proposals for service differentiation relying on distributed control. These architectures are based on the idea of modifying the backoff time

computation of the 802.11 standard to provide service differentiation, which is also the basis of our scheme.

In [19] the backoff time computation is modified by assigning shorter CWs to low delay real-time service. [20] and [21] propose the use of different CWs and different backoff increase parameters, respectively, for different priorities in data traffic. The fact that the parameters in [19], [20] and [21] are statically set makes the throughput received by a high quality station uncertain, as opposed to our proposal, in which the desired throughput is achieved by modifying dynamically the CW.

The Distributed Fair Scheduling (DFS) approach [22] proposes a dynamic algorithm for the backoff time computation in order to allocate bandwidth to the different stations proportionally to their weights. The main difference between the service provided by DFS and our approach is that DFS provides relative throughput guarantees, while our approach provides absolute guarantees. One drawback of DFS as compared to our approach is that in DFS each node has to monitor all transmitted packets and read the so-called finish tag of each packet. In addition, DFS requires the header format of 802.11 to be modified in order to include this finish tag in the packet header.

[23] provides relative priorities for delay and throughput in a multi-hop wireless network. This approach piggybacks scheduling information onto RTS/DATA packets and then uses this information to modify the computation of the backoff times. [23] has the same drawbacks commented for DFS, since it requires all nodes to monitor all transmitted packets in order to extract the scheduling information, and it requires the modification of the 802.11 header formats. Another drawback of [23] is that it does not provide backwards compatibility.

The Distributed QoS (D-QoS) approach [24] proposes the use of different CWs and Contention Offsets (COs) for different priority classes, in order to provide relative differentiation between classes. [24] mentions the possibility of dynamically computing the CW/CO values based on the monitored load. However, [24] does not propose any algorithm for this dynamic computation and provides only simulation results for statically set CWs/COs.

VII. Conclusions

In this paper we have proposed the ARME architecture for providing an Assured Rate Service in Wireless LAN, in line with the Assured Rate Service of DiffServ. The ARME architecture provides a scheduling comparable to the RIO scheduling of DiffServ for the wireline Internet.

The design goals of ARME have been to keep the MAC protocol fully distributed, to minimize the migration effort from the current standard, and to provide backwards compatibility. We argue that a fully distributed MAC protocol is more efficient and flexible than a centralized one. We believe that the fact that ARME only requires minor changes in the computation of the CW facilitates the migration from the 802.11 standard. Finally, the algorithm for the computation of the CW has been designed in such a way that 802.11 terminals behave as Best Effort terminals in the proposed architecture.

The simulations performed show that ARME provides Assured Rate terminals with its guaranteed throughput in normal circumstances, while the leftover bandwidth is shared equally between Best Effort and Assured Rate. Furthermore, starving Best Effort terminals is avoided in case of overload by trading off the bandwidth assurance of Assured Rate. Finally, simulations with TCP have shown that the algorithm is capable of overcoming the congestion control of TCP.

References

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

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," RFC 2475, December 1998.

[3] V. Jacobson, K. Nichols, and K. Poduri, "An Expedited Forwarding PHB," RFC 2598, June 1999.

[4] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group," RFC 2597, June 1999.

[5] V. Jacobson, K. Nichols, and K. Poduri, "The Virtual Wire Per-Domain Behavior," Internet draft, July 2000.

[6] B. Carpenter and K. Nichols, "A Bulk Handling Per-Domain Behavior for Differentiated Services," Internet draft, January 2001.

[7] N. Seddigh, B. Nandy, and J. Heinanen, "An Assured Rate Per-Domain Behavior for Differentiated Services," Internet draft, February 2001.

[8] M. Brunner, A. Banchs, S. Tartarelli, and H. Pan, "A one-to-any Assured Rate Per-Domain Behavior for Differentiated Services," Internet draft, February 2001.

[9] IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE Standard 802.11, June 1999.

[10] F.A. Tobagi and L. Kleinrock, "Packet Switching in Radio Channels: Part I - Carrier Sense Multiple-Access Modes and their throughput delay characteristics," *IEEE Trans. on Communications*, vol. 23, no. 12, pp. 1417–1433, 1975.

[11] F.A. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part II - the Hidden Terminal Problem in Carrier Sense Multiple-Access Modes and the Busy-Tone Solution," *IEEE Trans. on Communications*, vol. 23, no. 12, pp. 1417–1433, 1975.

[12] C. Fullmer and J. J. Garcia-Luna-Aceves, "Floor acquisition multiple access (fama) for packet radio networks," Computer Communication Review, vol. 25, (no. 4),(ACM SIGCOMM '95, Cambridge, MA, USA, 28 aug.-1 Sept. 1995) ACM, 1995.

[13] M. A. Visser and M. E. Zarki, "Voice and Data transmission over an 802.11 Wireless network," in *Proceeding of PIMRC*, Toronto, Canada, September 1995.

[14] D. D. Clark and W. Fang, "Explicit Allocation of Best Effort Packet Delivery Services," *IEEE/ACM Transactions on Networking*, vol. 6, no. 4, pp. 362–373, August 1998.

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

[16] J. Ibanez and K. Nichols, "Preliminary Simulation Evaluation of an Assured Service," draft-ibanez-diffserv-assured-eval-00.txt, Internet draft, August 1998.

[17] T. Nandagopal, S. Lu, and V. Bharghavan, "A Unified Architecture for the Design and Evaluation of Wireless Fair Queuing Algorithms," in *Proceedings of ACM MOBICOM*, Seattle, WA, August 1999.

[18] S. Lu, V. Bharghavan, and R. Srikant, "Fair Scheduling in Wireless Packet Networks," in *Proceedings of ACM SIGCOMM*, Cannes, France, August 1997.

[19] M. Barry, A. Veres, and A. T. Campbell, "Distributed Control Algorithms for Service Differentiation in Wireless Packet Networks," in *Proceedings of INFOCOM*, Anchorage, Alaska, April 2001.

[20] A. Ayyagari, Y. Bernet, and T. Moore, "IEEE 802.11 Quality of Service - White Paper," IEEE 802.11-00/028.

[21] A. Imad and C. Castelluccia, "Differentiation Mechanisms for IEEE 802.11," in *Proceedings of INFOCOM*, Anchorage, Alaska, April 2001.

[22] N. H. Vaidya, P. Bahl, and S. Gupta, "Distributed Fair Scheduling in Wireless LAN," in *Proceeding of MOBICOM*, Boston, MA, August 2000.

[23] V. Kanodia, C. Li, B. Sadeghi, A. Sabharwal, and E. Knightly, "Distributed Multi-Hop with Delay and Throughput Constraints," in *Proceeding of MOBICOM*, Rome,Italy, July 2001.

[24] G. Chesson, W. Diepstraten, D. Kitchin, H. Teunissen, and M. Wentink, "Baseline D-QoS Proposal," IEEE 802.11-00/399.