

Distributed Control Algorithms for Service Differentiation in Wireless Packet Networks

Michael Barry¹, Andrew T. Campbell², Andras Veres³

Abstract— This paper investigates differentiated services in wireless packet networks using a fully distributed approach that supports service differentiation, radio monitoring and admission control. Service differentiation is based on the IEEE 802.11 Distributed Coordination Function (DCF) originally designed to support best-effort data services. We extend the Distributed Coordination Function to provide service differentiation for delay sensitive and best-effort traffic. Two distributed estimation algorithms are proposed and analyzed. A *Virtual MAC (VMAC)* algorithm passively monitors the radio channel and estimates locally achievable service levels. The Virtual MAC estimates key MAC level statistics related to service quality such as delay, delay variation, packet collision and packet loss. We show the efficiency of the Virtual MAC algorithm and consider significantly overlapping cells and highly bursty traffic mixes. A *Virtual Source (VS)* algorithm utilizes the Virtual MAC to estimate application level service quality. The Virtual Source allows application parameters to be tuned in response to dynamic channel conditions based on “virtual delay curves”. We demonstrate through simulation that when these distributed virtual algorithms are applied to the admission control of the radio channel then a globally stable state can be maintained without the need for complex centralized radio resource management. Finally, we discuss a distributed service level management scheme that builds on the proposed algorithms to offer continuous service with handoff.

I. INTRODUCTION

In the past several years the Internet has started to penetrate the wireless world with the result that the emphasis in wireless communication will be more on TCP/IP based applications, in contrast to the current circuit switched voice. It is envisioned that TCP/IP will be the glue for all applications in future mobile environments, many of them requiring better than best-effort services. Wireless access may be considered just another hop in the communication path. It is desirable that the architecture supporting quality assurances follows the same principles in the wireless network as in the wireline Internet, assuring compatibility between the wireless and wireline parts.

There are two principal approaches to support better than best-effort services for Internet based services in a future wireless network. The first approach starts from the conventional circuit switched paradigm and extends it with datagram services. These systems are characterized by very strict control over both the wireline and wireless resources, motivated by the argument that such control, with complex and sophisticated mechanisms and protocols, is required to maintain good quality in the wireless environment [1][2][3].

Another increasingly popular approach is based on an important Internet design principle that mandates that only minimal control and signaling is viable, since only simple mechanisms can accommodate the diversity of applications in the Internet, let

alone unforeseen future wireless applications. A good example for such a wireless technology is the IEEE 802.11 standard [4], which in itself does not guarantee anything other than best-effort service for mobile hosts in the Distributed Coordination Function(DCF) mode. However, the IEEE 802.11 DCF enables the fast installation of simple wireless access networks, with minimum management and maintenance costs, and with virtually no requirement for cell planning. Similar distributed algorithms are analyzed and compared in [5] [6].

The IEEE 802.11 Point Coordination Function (PCF) is intended to support real-time services by using a central polling mechanism. This mechanism is not currently supported by most wireless cards, and the cooperation between PCF and DCF modes can lead to poor performance [7]. For real-time services, we argue that distributed control is more efficient and flexible than centralized control in the case of highly bursty traffic. Furthermore, the basic IEEE 802.11 DCF standard, which is not capable of supporting better than best-effort services, can in fact be extended to support service differentiation.

Creating a quality of service aware MAC is just one component of a differentiated services capable wireless network. In this paper, we discuss the components of such a network focussing on simplicity. These components include:

- a distributed DiffServ capable wireless MAC;
- a distributed radio resource monitoring mechanism;
- localized management of service level specifications and admission control; and
- fast mobility support in the access network.

Providing differentiated services in a mobile environment requires that the radio MAC supports some degree of separation between different types of services. We propose a modified IEEE 802.11 radio MAC algorithm for mobile hosts and base stations. The proposed MAC ensures that all packets sent by a mobile host are differentiated and, more importantly, that differentiation is effective among packets sent by other mobile hosts as well. Furthermore, IEEE 802.11 network cells may overlap significantly and service differentiation has to work across cells. This is achieved in a distributed manner without any central control over multiple cells.

Providing service differentiation solely at the radio interface is insufficient to enable predictable behavior for individual traffic types. In response to this, we develop *Virtual MAC (VMAC)* and *Virtual Source (VS)* algorithms that monitor the capability of the radio channel and passively estimate whether the channel can support new service demands (e.g., delay and loss) taking into account both local conditions and interference caused by external effects or overlapping cells. These “virtual algorithms” are passive and do not load the channel, avoiding further increases of load in potentially congested wireless networks. Based on the service quality estimations obtained from the vir-

¹Broadcom Eireann Research Limited, Dublin, Ireland; email mgb@broadcom.ie. ²COMET Group, Columbia University, New York, USA, email campbell@comet.columbia.edu. ³Traffic Lab, Ericsson Research, Budapest, Hungary; email Andras.Veres@ericsson.com.

M. Barry and A. Veres are Visiting Researchers with the COMET Group, Columbia University, New York, NY, e-mail: mgb@comet.columbia.edu and veres@comet.columbia.edu

tual monitoring algorithms, mobile nodes and base stations determine whether a new session with a particular service level requirement should be admitted or not. We show that if all nodes use passive monitoring and base their admission decisions accordingly, a globally stable state can be maintained even in multicell environments, without the need for cell planning or centralized control mechanisms.

The structure of the paper is as follows. In Section II, we briefly introduce the DCF mode of the IEEE 802.11 protocol. Section III discusses and analyses the achievable service differentiation by tuning the backoff algorithm of DCF. In Section IV, we present the Virtual MAC and Virtual Source algorithms, which estimate the achievable service quality level of the radio channel. Following this, we introduce the notion of “virtual delay curves” giving a number of examples for several traffic mixes in Section V. The use of virtual algorithms in a multicell network is analyzed using simulation in Section VI. Finally, we present a distributed service management mechanism for fast handoff in Section VII.

II. THE IEEE 802.11 MAC DCF PROTOCOL

The IEEE 802.11 MAC DCF protocol is a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol. In the DCF mode, a station must sense the medium before initiating the transmission of a packet. If the medium is sensed as being idle for a time interval greater than a Distributed Inter Frame Space (DIFS) then the station transmits the packet. Otherwise, the transmission is deferred and a backoff process is started. More specifically, the station computes a random value in the range of 0 to the so-called Contention Window (CW). A backoff time interval is computed using this random value: $T_{backoff} = Rand(0, CW) * T_{slot}$, where T_{slot} is the slot time [4]. This backoff interval is then used to initialize the backoff timer. This timer is decreased only when the medium is idle. The timer is frozen when another station is detected as transmitting. Each time the medium becomes idle for a period longer than DIFS, the backoff timer is periodically decremented by one for every slot-time.

As soon as the backoff timer expires, the station accesses the medium. A collision occurs when two or more stations start transmission in the same slot. An acknowledgement is used to notify the sending station that the transmitted frame has been successfully received.

If no acknowledgement is received, the station assumes that the frame transmission failed and schedules a retransmission by reentering the backoff process. To reduce the probability of collisions, after each unsuccessful transmission attempt, the Contention Window is doubled until a predefined maximum (CW_{max}) is reached. After a successful or unsuccessful frame transmission, if the station has frames queued for transmission, it executes a new backoff process.

To solve the hidden station problem the MAC protocol can use a short Request To Send (RTS) – Clear To Send (CTS) negotiation before sending a data packet. This reduces the collision probability for data packets but increases the protocol overhead.

The DCF mechanism has been investigated in numerous papers. The fairness of distributed control is investigated in [8]. A theoretical analysis of the protocol can be found in [9]. Analy-

sis and protocol enhancements for the distributed coordination function are presented in [10] [11] [12]. The shared medium access in case of multicell environment is analyzed using simulation in [13].

III. DIFFSERV ENABLED IEEE 802.11 RADIO MAC

A DiffServ enabled MAC must ensure that the available radio resources are shared among active users, while at the same time ensuring that the different traffic types receive service in a differentiated manner. The ideal radio MAC is adaptive and robust to both internal and external circumstances; that is, it offers effective protection for differentiated traffic classes against traffic fluctuations in lower classes. The MAC should also be robust to changes in the external environment, for example, growth of traffic in a cell must have a predictable and limited effect on the delay and loss experienced by all service classes in neighboring cells.

We argue that only decentralized and adaptive mechanisms can efficiently solve these problems. Distributed control of radio resources results in more productive use of those resources. Such distributed control allows mobile hosts within the same class to compete for radio resources and achieve acceptable fairness, while at the same time offering differentiated access to different service classes.

There have been several proposals for service differentiation in wireless networks. In [14] [3] several wireless scheduling algorithms are analyzed. It is shown that these algorithms approximate optimal fluid fair scheduling even in the presence of location-dependent error bursts. However, these mechanisms rely on centralized control and polling of backlogged mobile hosts. In addition, these algorithms are analyzed using short memory models (e.g., CBR, Poisson and MMPP), which have been shown to be inefficient when modeling real TCP/IP traffic [15]. In [16] a distributed solution to support real-time sources over IEEE 802.11 is introduced, by modifying the MAC for real-time sources to send short transmissions to gain priority. This method offers bounded delay, but has the disadvantage that it is optimized for isochronous sources, which can be a significant limitation for applications with variable data rates.

A. Discussion of Backoff Timers and Service Differentiation

The previous section describes how initial values used by the backoff procedure are determined using the CW parameter, which increases exponentially toward an upper bound as the backoff procedure is reset for a given transmission. In other words, the more transmission attempts for a given packet, the larger the CW becomes, and so the longer the time between transmission attempts.

Backoff times are set to a random value in the range $[0, CW] * T_{slot}$. After a collision, a new backoff time is chosen, but with an increased CW value. After every successful transmission, CW is reset to an initial value CW_{min} . We propose to support at least two service classes, high priority and best effort. Setting different CW_{min} values for each service class means that for two, or more, packets entering a backoff procedure at the same time, but with different CW_{min} values, the packet with the smaller value of CW is more likely to be transmitted first. Even if collisions occur, all MACs increase CW at the same

rate, and it is likely that the CW of high priority packets will remain lower than that of low priority packets with the result of experiencing smaller average delays. Intuitively, even during highly congested periods, all classes experience increased delays, but still in a differentiated manner.

By decreasing the maximum CW limit, CW_{max} , for a service class, the maximum backoff time can be limited during congestion. This limits the range of congestion control, thus we trade lower delay for increased collision probability, and eventually larger packet loss ratio. Nevertheless, we argue that for better than best-effort services it is preferable to drop a packet than to delay it excessively.

By setting different values of CW_{min} and CW_{max} for different traffic classes, different levels of service (i.e., delay and loss) can be achieved. There is, however, no explicit guarantee of the level of service differentiation. The next section explores the achievable service differentiation using these simple means of control. For simulation we use the ns-2 network simulator developed by the VINT Project [17] with the wireless extension produced by the MONARCH Group [18].

B. Simulation Results of the Modified MAC

Initially the degree of separation between high priority and best-effort traffic for different values of CW_{min} and CW_{max} is investigated for a fixed traffic mix consisting of delay sensitive voice sources and best-effort TCP transmissions. To demonstrate the operation of the modified MAC, the traffic mix consisted of 5 mobiles sending high priority voice traffic and 10 mobiles starting best effort greedy TCP connections. Voice traffic was modeled using an on/off source with exponentially distributed on and off periods of 300ms average each. Traffic was generated during the on periods at a rate of 32kbps with a packet size of 160 bytes, thus the inter-packet time was 40ms.

We ran a set of tests for this traffic mix with varying values for CW_{min} for both traffic classes. For high priority traffic the CW_{min} values varied between [8,32], the CW_{min} for best effort traffic varied between [32,128]. The value 32 is the proposed value by the standard [4] and applies to the case when only best effort traffic class is supported. We chose this value to be the delimiter between traffic classes during the test. The values chosen for high priority traffic range below this value. The value 8 is proposed by the standard as an absolute minimum. By using the values above 32 for best effort traffic class, the ranges do not overlap, and for all combinations it is assured that $CW_{min}^{highprio} \leq CW_{min}^{lowprio}$. Based on the intuitive discussion in the previous section, the maximum contention window of the high-priority class was lowered to $CW_{max}^{highprio} = 64$, while the upper limit for the low priority class was set to the recommended value of $CW_{max}^{lowprio} = 1024$.

In both intervals 5 values were chosen, spanning the range of CW_{min} values. Simulations were performed for all 5x5 combinations covering the whole plane. Packet delays were logged for both high and low priority traffic classes. Fig. 1 shows the summary of the simulations. The x axis corresponds to the CW_{min} of best-effort packets. It can be observed that increasing this value means larger delays for best-effort traffic and somewhat decreasing delays for real-time traffic. The dashed lines represent the delay of real-time sources. These are significantly

affected by the CW_{min} values. The delays of best-effort packets are not greatly affected by the value chosen for real-time sources, as shown by the straight line plots. In all combinations, apart from the trivial case where $CW_{min}^{highprio} = CW_{min}^{lowprio} = 32$, the streams in different traffic classes experienced differentiated delay. The experiment supports the argument that the delay differentiation can be increased by increasing the gap between $CW_{min}^{lowprio}$ and $CW_{min}^{highprio}$, i.e., decreasing $CW_{min}^{highprio}$ and increasing $CW_{min}^{lowprio}$.

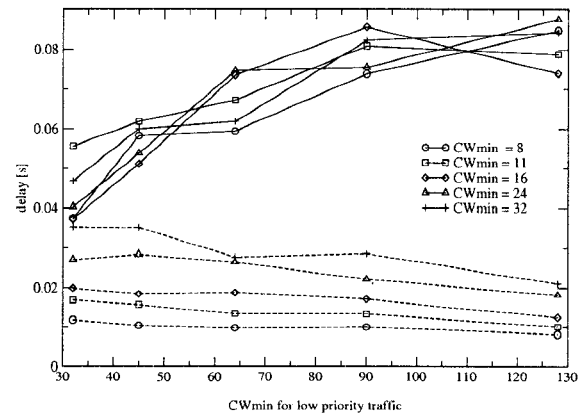


Fig. 1. Average delay experienced by voice and TCP flows for varying values of CW_{min} . The x axis represents the CW_{min} for best-effort traffic. Different symbols correspond to different CW_{min} for high-priority traffic.

The previous test demonstrated that effective service separation is possible by appropriately adjusting the backoff times through the contention window limits. However, it is still an open question whether this separation can be effectively maintained across a wide range of traffic loads for moderate to high congestion. In the next test, the robustness of service separation is investigated by simulating increasing levels of traffic up to the level of channel saturation.

In the simulation, the channel load is increased by adding a new voice, video (64kbps constant rate source) and TCP session periodically every 5 seconds. The voice and video sources use CW_{min} and CW_{max} values of 16 and 64, while the TCP traffic uses 128 and 1024, respectively. Fig. 2 shows the delay throughout the simulation for the three traffic types. We can observe that the delay increases for all types, but the delay separation is efficiently maintained from low load up until the channel is saturated.

For best-effort traffic the achievable throughput is of more importance than delay. Fig. 3 shows that the modified MAC enables the best-effort adaptive TCP traffic to utilize the free capacity leftover by high priority sources. It can be observed that even at the saturation point, the TCP traffic is not completely starved. This is due to the statistical and non-deterministic nature of service separation.

The modified MAC provides good service differentiation in terms of throughput and delay for a large range of high priority and best effort traffic mixes. A more dynamic traffic scenario is

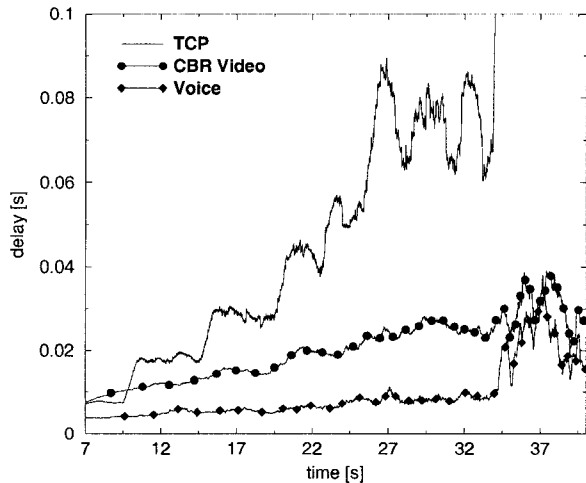


Fig. 2. Average delay experienced by gradually increasing the number of TCP and real time sources over time.

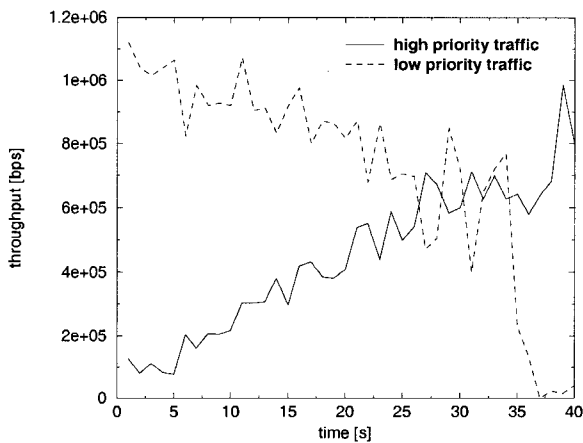


Fig. 3. Aggregate throughput of high priority and best effort traffic classes: number of TCP and real time sources increase over time.

investigated in Section VI.

IV. ESTIMATION OF AVAILABLE RESOURCES USING VIRTUAL SOURCE AND MAC ALGORITHMS

Many aspects of the wireless channel preclude exact control of resources, (e.g., channel fading or interference). The lack of cell planning and shared resources in the access network may result in densely packed base stations severely degrading the available capacity as perceived by neighboring base stations. The MAC described in the previous section ensures effective service differentiation even in cases of overlapping cells and high traffic loads. However, to support real-time services it is not sufficient to ensure that high priority traffic gets better service than best effort. In most cases applications require absolute and not relative service quality, (e.g., voice or video). If a mobile host realizes that the channel is unable to meet its delay and loss requirements, it can refrain from loading the channel further or reduce its application traffic demands, (e.g., by increasing compression).

In what follows, we present the Virtual MAC and Virtual Source algorithms that are capable of passively observing the radio channel. This passive monitoring capability allows a mobile host to evaluate the state of the channel, and estimate the level of service it would receive without actually loading the channel. In Section VI, we show that such an approach can lead to a stable state in multicell environments.

The channel can be in one of three states from the perspective of a new traffic flow: throughput limited, delay limited, or not congested. The “not congested” state means that the channel can probably serve the new traffic flow without severely degrading the channel state. Throughput limited state means that the channel is highly congested, and the new traffic flow would not be able to achieve the required throughput. The delay limited state means that although the channel may not be fully utilized, the delay experienced by the traffic source would probably exceed the requirements.

The free capacity of the channel can be estimated by measuring the idle channel time after DIFS on the radio channel. This time could be used for the transmission of data assuming ideal channel access and no collisions. We have found, that although this estimation is obviously overly optimistic, the channel usually becomes delay limited before becoming throughput limited.

Estimation of achievable delay is performed using Virtual Source and Virtual MAC algorithms. These algorithms operate in parallel to the real applications and MAC layer in the host and estimate the service level of the channel. They generate virtual data packets and MAC traffic emulating the behavior of real source and MAC mechanisms. However, no actual transmission is made over the air interface.

The VS algorithm consists of a Virtual Application, an interface queue and a Virtual MAC. The Virtual Application generates virtual packets as a real application would do (e.g., generating virtual voice packets at a constant rate). Packets are time-stamped and placed into a virtual buffer. Scheduling on the radio channel by the wireless MAC is performed in the same way as real packets, which means channel testing and random back-off is performed as necessary. The difference arises when the MAC decides to send the virtual packet. Unlike the case of real packets, no packet is transmitted at this point. Rather, the Virtual MAC algorithm estimates the probability of collision if a virtual packet were sent.

To make the algorithm conservative, collision is “detected” whenever any other mobile host chooses the same slot for transmission, (i.e., the channel is occupied by any mobile host within the same slot time). In this case the Virtual MAC enters a back-off procedure, as a real MAC would do after a collision. In a real MAC, collision detection is realized using an RTS timer. If no CTS has been received before this timer expires, after the transmission of an RTS, then a collision is assumed and the packet must be retransmitted. A real MAC would begin a backoff procedure at this point of time. The Virtual MAC does not detect collisions in this way. Rather, it decides that a collision would have happened if a transmission occurs in the timeslot determined by its backoff algorithm. In other words, the Virtual MAC detects “virtual collisions” immediately and not through using a timer. Thus, the VMAC enters the backoff procedure after a delay equal to that of an RTS timer in a real MAC.

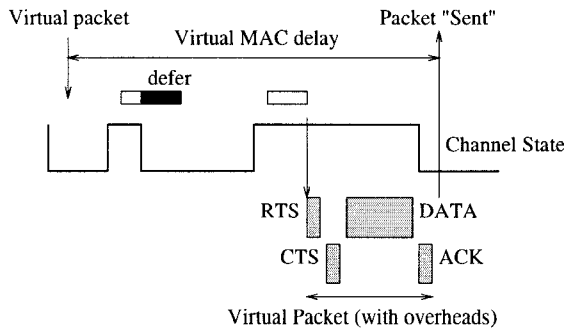


Fig. 4. Virtual MAC.

If no collision occurs, the MAC delay is estimated by the total defer time accumulated plus the time to fully acknowledge the packet (e.g., if RTS/CTS is enabled it is $d = t_{defer} + t_{RTS} + t_{CTS} + t_{packet} + t_{ACK} + 3t_{SIFS} + 3\tau$ where τ is an estimate of the maximum propagation delay). The mechanism is illustrated in Fig. 4.

All other aspects of a real MAC are emulated: maximum number of retries, adjusting the contention window, etc. After a successful “transmission”, the VMAC serves the next packet from the buffer, if there is one.

The VMAC and VS algorithms are designed to continuously keep track of the health of the channel. The VMAC measures virtual packet delays, packet losses and collisions at the MAC level. The application level VS estimates are obtained using mechanisms provided by the VMAC. The application level delay, for example, is estimated by examining the timestamp of a virtual packet successfully served by the VMAC. An application level loss event is triggered when the VMAC reaches the limit of retransmission attempts of a virtual packet. Estimates provided by the VMAC and VS algorithms can be used by a mobile host prior to the actual transmission of data. Because the virtual algorithms do not require high processing capacity, and obviously do not load the channel, they may run continuously and not only when a service request arrives.

Fig 5 shows a test of the efficiency of the Virtual MAC algorithm. The figure shows the simulated and the VMAC estimated delays experienced by a new real-time voice source for an increasing number of homogeneous voice sources. The estimation is precise across the whole range of traffic loads, most importantly in the saturation region. Thus, it is suitable for evaluating the admissible capacity of the channel for real time traffic.

Fig. 6 shows the results of a more complex test, where the voice traffic is mixed with an increasing number of “Web sources”. The Web sources are modeled by short TCP file transfers, where the file sizes are drawn from a Pareto distribution with mean file size of 10kbytes and shape parameter 1.2. The length of the silent period between two downloads is also Pareto with the same shape parameter and mean delay of 10s. This creates a highly bursty background data traffic load with multiple time-scale fluctuations [19] [20] [21]. This TCP load is itself sufficient to saturate the channel.

The figure shows two scenarios. In the first, the voice source is not prioritized over the data sources. In the second, the MAC algorithm is modified, as discussed in the previous section for

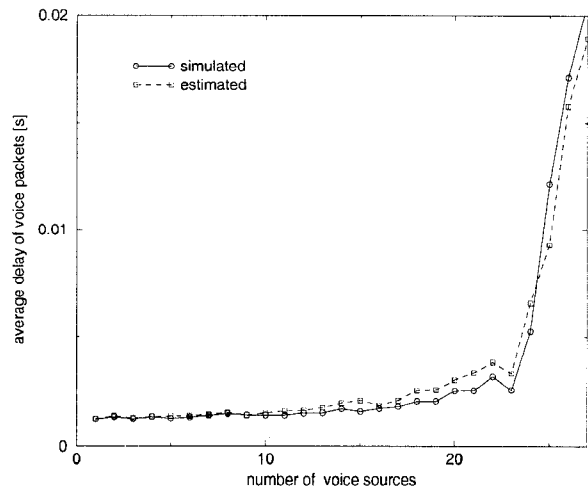


Fig. 5. Virtual and simulated average MAC delay of a new voice source versus the number of active voice sources.

the voice source. The results show that the delay is efficiently estimated by the Virtual Source algorithm. In both cases, the estimation is conservative and the mean delay is about 1-2 ms greater than the result obtained by simulation. Another important observation is that priority for voice provides significantly smaller and smoother delay and delay variation values in the case of highly bursty data traffic. Without modifying the MAC for voice, the voice packets have to compete with data packets, which, since the data traffic is much more bursty, does not only increase the voice packets’ delay but also increases the delay variance, as shown in Fig. 6.

V. VIRTUAL DELAY CURVES

The delay experienced by an application is made up of several components: packetization delay, delay accumulated in the buffers and the delay caused by the wireless MAC. The Virtual MAC algorithm estimates the MAC delay, which is the time needed from receiving a packet by the MAC until the packet is acknowledged at the MAC layer and the next packet is serviced. The application may experience significantly higher delays than the MAC delay, as there is an interface queue between the application and the MAC layer to smooth out bursts caused by the MAC algorithm and the application. The total delay of packetization, buffering and medium access is estimated by the Virtual Source algorithm, which emulates all these components in concert.

The application delay depends on several factors. Certain factors depend on the application such as packet size, packet rate, others depend on the load of the channel. The VS algorithm, which monitors the channel continuously, estimates the application performance taking into account these factors. Thus, the VS algorithm can be used to find the optimal parameters for the best application performance.

Intuitively, at the same data bitrate, the application delay can be reduced by increasing the packet rate, since it reduces the packetization delay. In contrast, higher packet rates load the radio channel more. Higher rates cause more collisions, increasing the average contention window. This eventually leads

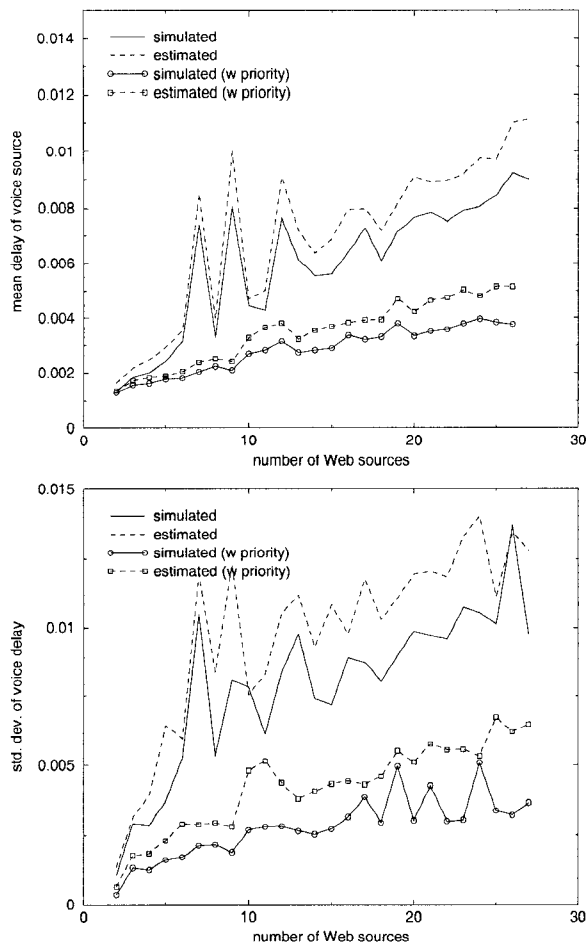


Fig. 6. Average delay (top) and delay variation (bottom) of a new voice source obtained by simulation and from the virtual MAC algorithm, versus number of Web sources with and without priority for voice traffic.

to larger MAC delays. In addition, higher packet rates mean smaller data packets, which results in larger protocol overhead, (i.e., larger load on the radio channel). Thus, even at the same application bitrate, there is a tradeoff between packetization delay and MAC delay.

Denote the function $d(p_{rate})$ as the *virtual delay curve* of an application, where p_{rate} is the packet inter-arrival time of the application, e.g., $p_{rate} = 0.02$ packets per second for voice, but the data bitrate is kept constant, i.e., $p_{size} * p_{rate} = const$ (where p_{size} is the size of the application level packet). The virtual delay curve at p_{rate} gives the average delay of virtual packets if the VS algorithm generates packets at the rate of p_{rate} . The mobile host or the base station runs VS algorithms with several p_{rate} values in parallel. From the virtual packet delays obtained from the VS algorithm, the delay curve can be constructed. Similarly, we can define the *virtual delay variance curve* $v(p_{rate})$ which calculates the virtual delay variances, respectively. Based on the delay curve, the mobile host or the base station can choose the optimal packet rate and packet size so that the application experiences minimum delay and delay variance.

A standard set of traffic types should be used with the VS algorithms, each of which corresponds to a common voice, video

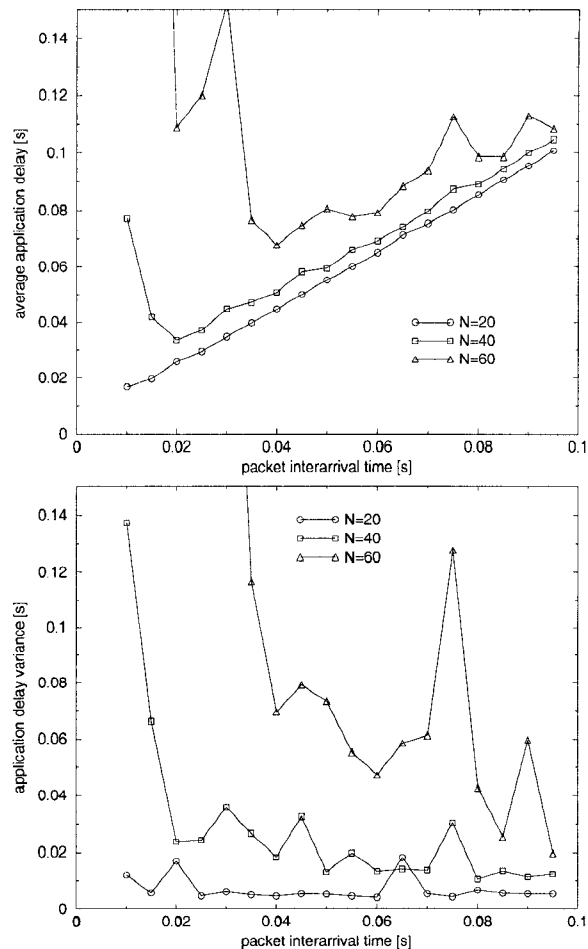


Fig. 7. Virtual delay and delay variance curves at several radio channel loads, N denotes the number of background Web sessions. Web traffic uses $CW_{min} = 31$ and the virtual algorithm uses $CW_{min} = 8$.

or data stream, (e.g. 32kbit 10ms inter-packet time, or 128 kbps data stream). Where an application does not conform to the standard types, a conservative match could be used. Otherwise the application must rely on the output of the VMAC alone to estimate the quality level.

Fig. 7 shows the virtual delay and variance curves of a virtual voice source at several background traffic loads. It can be observed that in the case of low background traffic ($N = 20$ "Web" sessions) the delay curve increases monotonically, which means that the best end-to-end delay can be achieved if the packet rate is high and the application sends small packets. The estimation of delay variance appears to be constant. As the background load increases, the MAC delay increases and the optimum is not at the highest rate but at about 20 ms. The delay variance also decreases as the inter-packet times increase.

VI. DISTRIBUTED ADMISSION CONTROL ALGORITHM IN A MULTICELL ENVIRONMENT

The previous sections demonstrated that the modified MAC and the virtual algorithms are capable of providing predictable performance for real-time sources. Based on the VS and VMAC results, an admission control (AC) algorithm can determine

whether the channel can support a new traffic stream or not. This algorithm compares the results of the VS and VMAC with the service requirements and admits or rejects a new session accordingly. The natural location of the AC algorithm is at the base station that the mobile host is associated with. If all the base stations in the area execute the same algorithm, a globally stable state can be maintained. This is true even if cell areas overlap and share radio resources.

Since the radio channel properties may be different at the mobile host than at the base station, it is preferable that the mobile host also executes a VS algorithm to ensure that the service quality will be met for the new stream. Admission is granted if both VS algorithms at the mobile and base station admit the new request. In what follows, we investigate this concept through simulation of a complex configuration with multiple base stations, random topology and random traffic events. The aim is to test how the modified MAC and VS algorithms perform in the presence of a highly dynamic real-time and non real-time traffic mix, where the radio channel is dynamically shared among traffic streams between mobile nodes and base stations.

Independent Virtual Source algorithms running in all base stations continuously monitor the radio channel. Admission control is applied for the delay sensitive voice sessions. In contrast, no admission control is applied to Web traffic. In general, we do not apply admission control techniques to low priority traffic. We have already shown how a differentiated service MAC enforces a clear separation between high and low priority traffic, even at the saturation point. Furthermore, in the current best-effort Internet it would be very difficult to introduce any form of admission control for best-effort traffic, as this would require changes to existing applications, new protocols, etc.

Ten base stations are placed randomly on a 400m by 400m rectangular area with their coverage areas significantly overlapping. One hundred mobile nodes are placed randomly in the coverage area. Every mobile host is associated with the nearest base station. Half of the mobile hosts randomly generate Web sessions and the other half randomly generate voice sessions. The length of the voice sessions and the inter-arrival times between connection requests are exponentially distributed. The average call length is 30s. Upon completion of a call, a mobile attempts a new call after an average waiting period of 10s. When the estimated average delay exceeds 10 ms, new voice sessions are rejected from service. If accepted, the voice packets use the modified MAC algorithm with $cw_{min} = 32$ slots and $cw_{max} = 64$ slots, while the Web sessions use values 64 and 1024, respectively.

Fig. 8 shows the total TCP and voice traffic rates in the entire coverage. After an initial startup, the aggregate voice rate settles around a stable throughput, while the TCP traffic shows high levels of burstiness.

Fig. 9 shows the delay estimations by the VS algorithms running in base stations. It can be observed that the delay estimate remains below the admission target most of the time for most base stations. However, the estimated delay is significantly different at a few base stations, where, the estimated delay reaches 10 ms for long durations. These base stations do not accept voice traffic during these periods. In contrast, other base stations are continuously in the accept state. This is due to the

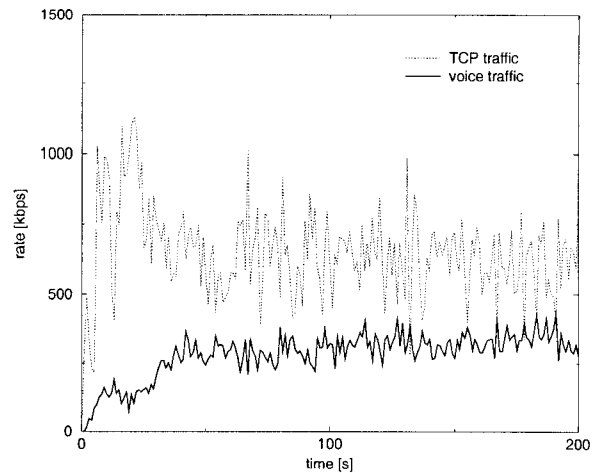


Fig. 8. Aggregate rates of TCP and voice traffic in the entire service area.

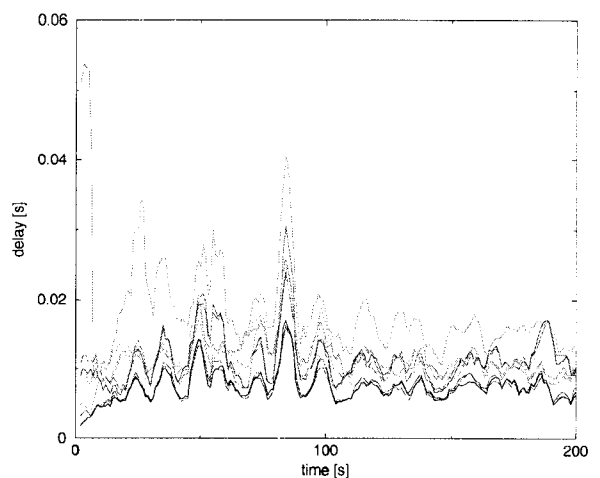


Fig. 9. Estimated delays by VS algorithms running at base stations.

overlapping of cells and the shared radio channel.

Fig. 10 shows the actual delays experienced by admitted voice sessions. The empirical distribution of voice packet delays is shown in Fig. 11.

The low delays experienced by admitted sessions indicate that the overall channel state is efficiently controlled by the distributed monitoring and admission control algorithms, even in the presence of highly dynamic TCP traffic.

VII. DISTRIBUTED SERVICE LEVEL MANAGEMENT IN A WIRELESS ACCESS NETWORK

As the mobile host moves in the coverage area and performs handoff between base stations, it is necessary to conduct admission control repeatedly. In the previous sections we presented a solution for admission control that is fully distributed, based exclusively on local decision making. However, to perform the admission control decision, the base station has to know the negotiated Service Level Agreement (SLA) between the mobile host and the network, which is agreed at the beginning of service. Thus, the SLA context has to be transferred to the new base station during handoff to perform admission control. This

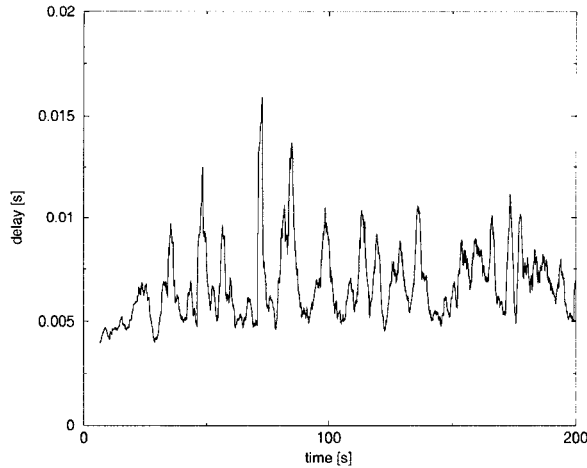


Fig. 10. Measured delays of voice packets (running average).

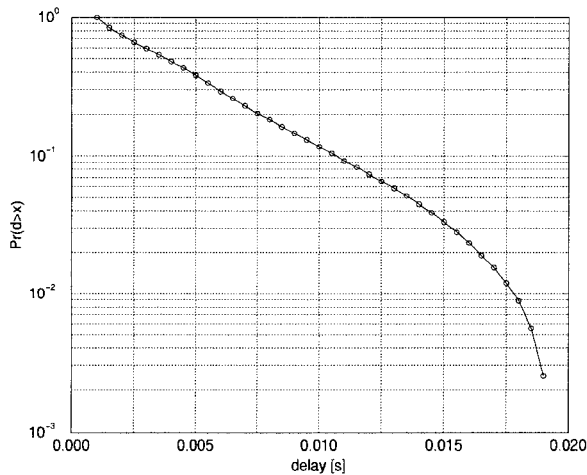


Fig. 11. Delay distribution of voice packets.

approach would require a centralized database and would not offer fast and localized decision making and continuous service in micro-mobility networks. In contrast, we propose a fast and fully distributed solution to this problem that comprises two steps. The first step is when the mobile authenticates with the network for the first time or wants to renegotiate its service (e.g., it starts or ends a voice session). This negotiation probably takes place between a centralized resource control agent and the mobile host. If the agent authenticates the host and decides that the request can be granted, it passes an *SLA token* to the mobile host, as illustrated in Fig. 12.

The token contains all the necessary details associated with the level of service, traffic specification (e.g., token bucket parameters or peak rate) and may contain an expiration time for security reasons. The token is encrypted and has no meaning to the mobile host but any base station in the access network can decrypt it using an access network specific secret key. When a mobile performs handoff within the access network, it only has to pass the *SLA token* to the new base station, which performs admission control locally. In this manner, no communication is needed between the mobile host and a centralized agent. This

allows admission control to be fully distributed and very fast, ensuring continuous service for mobile hosts during handoff. Fig. 12 illustrates network level and localized admission procedures in the access network.

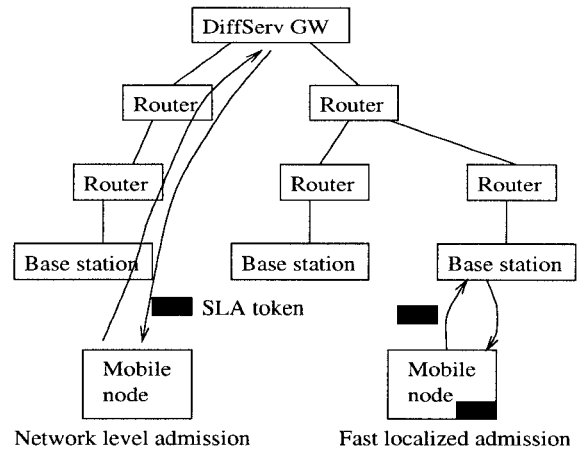


Fig. 12. Fast and localized admission control using encrypted *SLA tokens*.

VIII. CONCLUSION

This paper has shown how service differentiation can be provided in a mobile access network in a fully distributed manner with minimal control. By manipulating the contention window limits of the IEEE 802.11 DCF mode it is possible to provide service differentiation at the radio MAC layer. The proposed MAC provides good delay and throughput separation for best effort and high priority traffic for a range of traffic mixes and channel loads.

We have proposed two passive radio channel monitoring algorithms. By emulating MAC (Virtual MAC) and application (Virtual Source) mechanisms, these algorithms can estimate the achievable level of service without actually loading the channel. The notion of virtual delay curves has been introduced in relation to the virtual algorithms. Delay curves enable an application to tune its traffic parameters to match the dynamic characteristics of the radio channel in an efficient manner.

We have demonstrated through simulation that the modified MAC together with a distributed admission control algorithm can maintain a globally stable state in a micro-cellular environment even if cell areas overlap and the radio channel is shared.

We have outlined how these mechanisms can be incorporated into a broader wireless DiffServ network that supports a scheme for distributed service level management, which offers continuous service levels while supporting handoff.

Finally, we are currently building a wireless DiffServ testbed using Cellular IP [22] for mobility management. This testbed will include the modified MAC and virtual control algorithms and will provide support for service level agreements with fast handoff. Results from this phase of our work will be the subject of a future publication.

IX. ACKNOWLEDGEMENTS

Michael Barry and Andras Veres would like to thank Broadcom Eireann Research and Ericsson Research, respectively, for

supporting this work during their stay with COMET Group as Visiting Researchers. The National Science Foundation (NSF) under the WIRELESS TECHNOLOGY Award ANI-9979439 has supported Andrew T. Campbell's work on this project. Finally, the authors would like to thank Professor Mischa Schwartz for his great input.

REFERENCES

- [1] M. Mouly, M. B. Pautet, "The GSM System for Mobile Communications", *Telecom Pub*, June 1992
- [2] ETSI, "Digital Cellular Telecommunications System General Packet Radio Services: Service Description EN 301 344 v6.3.2", July 1999
- [3] T. Nandagopal, S. Lu, V. Bharghavan, "A Unified Architecture for the Design and Evaluation of Wireless Fair Queueing Algorithms", *Proceedings of MOBICOM*, Seattle, WA, Aug. 1999
- [4] IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", *IEEE Standard 802.11*, June 1999
- [5] V. Bharghavan, "Performance Analysis of a Medium Access Protocol for Wireless Packet Networks", *IEEE Performance and Dependability Symposium*, Raleigh, NC, Aug. 1998
- [6] V. Bharghavan, A. Demers, S. Shenker, L. Zhang, "MACAW: A Media Access Protocol for Wireless LANs", *Proceedings of ACM SIGCOMM*, London, UK, Sept. 1994
- [7] M. A. Visser, M. E. Zarki, "Voice and Data Transmission over an 802.11 Wireless Network", *Proceedings of PIMRC*, Toronto, Canada, Sept. 1995
- [8] T. Nandagopal, T.-E. Kim, X. Gao, V. Bharghavan, "Achieving MAC Layer Fairness in Wireless Packet Networks", *Proceedings of MOBICOM*, Boston, MA, Aug. 2000
- [9] F. Cali, M. Conti, E. Gregori, "IEEE 802.11 Wireless LAN: Capacity Analysis and Protocol Enhancement", *Proceedings of IEEE INFOCOM*, Apr. 1998.
- [10] G. Bianchi, L. Fratta, M. Oliveri, "Performance Evaluation and Enhancement of the CSMA/CA MAC Protocol for 802.11 Wireless LANs", *Proceedings of PIMRC*, Taipei, Taiwan, Oct. 1996
- [11] J. Weinmiller, M. Schlager, A. Festag, A. Wolisz, "Performance Study of Access Control in Wireless LANs - IEEE 802.11 DFWMAC and ETSI RES 10 HIPERLAN", *Mobile Networks and Applications*, Vol. 2, pp. 55-67, 1997
- [12] J. Weinmiller, H. Woesner, J. P. Ebert, A. Wolisz, "Analyzing and Improving the 802.11-MAC Protocol for Wireless LANs", *Proceedings of MAS-COT*, San Jose, CA, Feb. 1995
- [13] C. C. Chow, V. C. M. Leung, "Performance of IEEE 802.11 Medium Access Control Protocol over a Wireless Local Area Network with Distributed Radio Bridges", *Proceedings of WCNC*, New Orleans, LA, Sep. 1999
- [14] S. Lu, V. Bharghavan, R. Srikant, "Fair Scheduling in Wireless Packet Networks", *Proceedings of ACM SIGCOMM*, Cannes, France, 1997
- [15] V. Paxson and S. Floyd, "Wide Area Traffic: The Failure of Poisson Modeling", *IEEE/ACM Transactions on Networking*, vol. 3, pp. 226-244, June 1995.
- [16] J. L. Sobrinho, A. S. Krishnakumar, "Real-Time Traffic over the IEEE 802.11 Medium Access Control Layer", *Bell Labs Technical Journal*, Autumn 1996
- [17] L. Breslau, D. Estrin, et al, "Advances in Network Simulation", *IEEE Computer Magazine*, May 2000
- [18] CMU MONARCH Project, www.monarch.cs.cmu.edu
- [19] W. E. Leland, M. S. Taqqu, W. Willinger and D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic", *ACM SIGCOMM '93*, San Francisco, CA, USA, Sept. 1993
- [20] M. E. Crovella and A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes", in *Proc. ACM SIGMETRICS*, pp. 160-169, May 1996.
- [21] M. S. Taqqu, W. Willinger and R. Sherman, "Proof of a Fundamental Result in Self-Similar Traffic Modeling", *Computer Communication Review*, 27 (1997) 5-23.
- [22] A. Valko, "Cellular IP - A New Approach to Internet Host Mobility", *Computer Communication Review*, Vol. 29, pp. 50-65, Jan. 1999