

Supporting Service Differentiation in Wireless Packet Networks Using Distributed Control

Andras Veres, Andrew T. Campbell, Michael Barry, *Member, IEEE*, and Li-Hsiang Sun, *Student Member, IEEE*

Abstract—This paper investigates differentiated services in wireless packet networks using a fully distributed approach that supports service differentiation, radio monitoring, and admission control. While our proposal is generally applicable to distributed wireless access schemes, we design, implement, and evaluate our framework within the context of existing wireless technology. Service differentiation is based on the IEEE 802.11 Distributed Coordination Function (DCF) originally designed to support best-effort data services. We analyze the delay experienced by a mobile host implementing the IEEE 802.11 DCF and derive a closed-form formula. We then extend the DCF to provide service differentiation for delay-sensitive and best-effort traffic based on the results from the analysis. Two distributed estimation algorithms are proposed. These algorithms are evaluated using simulation, analysis, and experimentation. A Virtual MAC (VMAC) algorithm passively monitors the radio channel and estimates locally achievable service levels. The VMAC 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 VMAC algorithm through simulation and consider significantly overlapping cells and highly bursty traffic mixes. In addition, we implement and evaluate the VMAC in an experimental differentiated services wireless testbed. A Virtual Source (VS) algorithm utilizes the VMAC to estimate application-level service quality. The VS 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.

Index Terms—Distributed algorithms, quality of service, wireless packet networks.

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 toward TCP/IP-based applications rather than 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. Therefore, 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 begins with the conventional circuit switched paradigm and extends it with datagram services. These systems are characterized by strict control over both the wireline and wireless resources, motivated by the argument that such control, with complex and sophisticated mechanisms and protocols, is necessary to maintain good quality in the wireless environment [1]–[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. The Differentiated Services concept of the IETF [9] follows this design philosophy. By definition, it only specifies per-hop-behaviors instead of end-to-end services or protocols. The Differentiated Services architecture is envisioned to span the whole end-to-end path (e.g., from a server to a mobile user), thus the wireless hop has to be compatible with the Differentiated Services model as well. A good example for such a wireless technology is the IEEE 802.11 Distributed Coordination Function (DCF) standard [4], which is compatible with the current best-effort service model of the Internet. 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].

In the case of *ad hoc* wireless networks, there is no notion of a central entity. The dynamic nature of *ad hoc* networks makes it very difficult to dynamically assign a central controller and maintain connection, reservation, and scheduling states, not to mention the difficulty of handling overlapping coverage areas, in which case nearby mobile hosts need to discover and negotiate resources. Instead of introducing complex layer-two signaling, distributed algorithms attempt to solve these problems in a more straightforward, although possibly less radio-efficient, way.

In this paper, we propose a set of algorithms that form a fully distributed wireless differentiated services network based on:

- a distributed, differentiated services-capable MAC;
- a distributed radio resource monitoring mechanism;
- service quality estimation;
- distributed traffic and admission control.

Manuscript received December 1, 2000; revised June 1, 2001. The work of A. T. Campbell was supported by the National Science Foundation under the WIRELESS TECHNOLOGY Award ANI-9979439.

A. Veres is with the COMET Group, Columbia University, New York, NY 10027–6699 USA, on leave from Traffic Lab, Ericsson Research, H-1037 Budapest, Hungary (e-mail: veres@comet.columbia.edu).

A. T. Campbell and L.-H. Sun are with the COMET Group, Columbia University, New York, NY 10027–6699 USA (e-mail: campbell@comet.columbia.edu).

M. Barry is with the COMET Group, Columbia University, New York, NY 10027–6699 USA, on leave from Broadcom Eireann Research Limited, Dublin 2, U.K. (e-mail: mgb@broadcom.ie).

Publisher Item Identifier S 0733-8716(01)08486-4.

Each of these components performs a well-defined task and can be implemented in a fully distributed manner, without the need for a centralized controller. While our framework is generally applicable to distributed wireless access schemes, we design, implement, and evaluate our framework within the context of existing wireless technology. Service differentiation is based on the IEEE 802.11 DCF. Supporting better than best-effort service over such a shared wireless channel using distributed control algorithms presents a number of challenges, however.

The first challenge relates to the difficulty in providing service differentiation at the distributed wireless MAC layer. The impact of packet collisions, hidden terminals, fading, and interference suggests that such a radio environment lends itself more to soft service assurances rather than deterministic ones. In this work, we take our lead from this observation and attempt to quantify the level of assurance and service differentiation that can be delivered to wireless Internet applications. This means that under such a regime quality measures can only be probabilistically guaranteed where relative quality differentiation can be assured for different service classes.

Providing differentiated services in this manner 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.

Providing service differentiation solely at the radio interface is insufficient to enable predictable behavior for individual traffic types, however. This leads to our next challenge. Network cells may overlap significantly and service differentiation has to be maintained across cells. The probabilistic assurances offered by such a wireless differentiated services MAC itself cannot ensure that traffic levels experienced by a mobile host are not only relatively better but kept within some absolute limit for acceptable application quality. We address this challenge by proposing a distributed solution without the need for any central control over multiple cells. In particular, we propose a distributed traffic control algorithm, which maintains the traffic load such that the relative assurances offered by a differentiated services MAC can also meet the absolute limits required by the applications using better than best-effort services.

In response to these challenges, we develop the 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. The VMAC channel-monitoring capability is capable of collecting information about all transmissions in the proximity of a mobile host. Mobile hosts utilize this information to estimate the quality experienced by other mobiles. The difficulty of estimation in this environment is that there is little relationship between the monitored channel load and the delay or loss statistics of the channel. The VMAC and VS algorithms, which, based on the information provided by a passive channel monitor, can efficiently estimate the necessary quality metrics for different traffic classes. 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 virtual monitoring algorithms, mobile hosts and base stations determine whether a new session with a particular service level requirement should be admitted or not. In this paper, we simplify traffic control and propose an admission control solution that simply accepts or rejects real-time sessions. Admission is granted if the average delay estimated by the VS algorithm in the last time period falls within a certain delay limit. 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.

In this paper, we present the design, implementation, and evaluation of our framework. The principles that underpin our distributed approach are based on minimal control and signaling. While our implementation is evaluated within the context of IEEE 802.11, the algorithms that support service differentiation, radio monitoring, and admission control are more generally applicable. We recognize that such an approach can only deliver softer assurances in comparison to more tightly coupled control systems. We argue, however, that distributed control is more scalable (i.e., provides minimum coupling between architectural components), extensible (i.e., one component can be replaced or improved without the need to change other system components), and flexible (i.e., in accommodating new and diverse needs of applications).

The structure of the paper is as follows. Section II presents the related work on wireless service differentiation and distributed control. Section III discusses and analyzes the achievable service differentiation using a distributed approach. We analyze the delay experienced by a mobile host implementing the IEEE 802.11 DCF and derive a closed-form formula. We then extend DCF with the capability to tune and set the backoff mechanisms to provide service differentiation for delay sensitive and best-effort traffic based on the results from the analysis. In Section IV, we introduce the VMAC, which 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 VMAC algorithm through simulation, and in Section V we implement and evaluate the VMAC in an experimental differentiated services wireless testbed. In Section VI, we present the VS algorithm, which utilizes the VMAC to estimate application-level service quality. The VS allows application parameters to be tuned in response to dynamic channel conditions based on "virtual delay curves." In Section VII, we demonstrate through simulation that, when these distributed virtual algorithms are applied to the admission control of the radio channel, a globally stable state can be achieved. Finally, we present some concluding remarks and discuss future work.

II. RELATED WORK

Effective wireless MAC protocols must find a good balance between the added complexity of offering service guarantees for multiple service classes, the most efficient use of available resources, and the ability to react promptly to failed transmissions [7]. A number of MACs intended for third-generation protocols

are analyzed in [7], some of which offer probabilistic guarantees. In general, these MAC protocols and wireless algorithms rely on centralized control.

In [3] and [10], a number of wireless scheduling algorithms are analyzed, several of which approximate optimal fluid fair scheduling even in the presence of location-dependent error bursts. However, these mechanisms rely on centralized control and the polling of backlogged mobile hosts. 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 [11].

A distributed architecture to support weighted rate differentiation among flows is introduced in [26]. This proposal works on an end-to-end manner, where the end hosts adjust their rate using the Additive Increase Multiplicative Decrease (AIMD) algorithm. Instead of using packet loss, the AIMD actions are based on the observed end-to-end packet separation, which is treated as a sign of congestion. The algorithm works over low-bandwidth links and assumes that sources are greedy.

The IEEE 802.11 Point Coordination Function (PCF) is intended to support real-time services by using a centralized polling mechanism. This mechanism is not supported by most current wireless cards, however. In addition, cooperation between PCF and DCF modes leads to poor performance [12].

We argue that distributed control for supporting real-time services is more flexible than centralized control in the case of highly bursty traffic. We argue that 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. The DCF mechanism of IEEE 802.11 has been investigated in numerous papers.

In [21], a distributed solution for the support of real-time sources over IEEE 802.11 is discussed, which modifies the MAC to send short transmissions to gain priority for real-time service. It is shown that this approach is capable of offering bounded delay. One disadvantage of the design [21] is that it is optimized to meet the service needs of isochronous traffic sources, which can be a significant limitation for applications with variable data rates.

The fairness of distributed control is investigated in [15] and [16]. Both papers suggest distributed algorithms for rate-based service differentiation. Both papers solve the problem of throughput fairness. However, these contributions do not analyze the level of delay differentiation.

A theoretic analysis of the DCF protocol can be found in [17]. Analysis and protocol enhancements for the DCF are presented in [18]–[20]. Shared medium access for multicell environments is analyzed using simulation in [8].

III. DISTRIBUTED DIFFSERV ENABLED WIRELESS MAC

Providing differentiated services in a mobile environment requires that the radio MAC supports some degree of separation between different types of services. This separation is based on the DiffServ field in IP packets [9]. A “DiffServ enabled MAC” has to ensure that available radio resources are shared among active users, while at the same time ensuring that different traffic types receive service in a differentiated manner. The ideal radio

MAC is adaptive and robust to both internal and external dynamics, that is, it offers effective protection for the 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 decentralized and adaptive mechanisms can more efficiently solve these problems in comparison to centralized ones. Distributed control of the radio resources may result in more productive use of radio resources: by productive we mean a better performance/complexity ratio. Distributing control of the radio resources 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.

We propose a simple modification of the IEEE 802.11 radio MAC algorithm that runs in mobile hosts and base stations. The proposed MAC ensures not only that packets sent by the host itself are differentiated, but more importantly, that differentiation is effective among packets sent by other mobile hosts as well. Furthermore, IEEE 802.11 network cells may overlap significantly where service differentiation has to be maintained across cells. We show how this can be achieved in a distributed manner without any central control over multiple cells.

A. 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 mobile host transmits the packet. Otherwise, transmission is deferred and a backoff process is entered. 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: $T_{\text{backoff}} = \text{Rand}(0, \text{CW}) * T_{\text{slot}}$, 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 once every slot time.

As soon as the backoff timer expires, the mobile host accesses the medium. A collision occurs when two or more mobile hosts start transmission simultaneously in the same slot. An acknowledgment is used to notify the sending station that the transmitted frame has been successfully received. If an acknowledgment is not received, the station assumes that the frame was not received successfully and schedules a retransmission, reentering the backoff process. To reduce the probability of collisions, after each unsuccessful transmission attempt, the CW is doubled until a predefined maximum (CW_{max}) is reached. After a successful or unsuccessful frame transmission, if the station still has frames queued for transmission, it must execute a new backoff process.

To deal with the hidden terminal 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.

In [15] and [16], more sophisticated distributed algorithms have been proposed that address the inefficiency of the original CSMA/CA algorithm for providing fair access. These methods can replace our simple DiffServ MAC algorithm in the distributed architecture. In this case, the virtual algorithms discussed later in this paper would have to be modified accordingly. Nevertheless, the advantage of our algorithm is that it is closer to the original IEEE 802.11 standard and can be easily implemented in existing IEEE 802.11 cards.

B. Delay Analysis of the Distributed Coordination Function

Previous work has analyzed the IEEE 802.11 DCF mode from several different perspectives, including fairness, throughput, and the effect of hidden terminals. We are interested, however, in analyzing the kind of delay guarantees that can be achieved using DCF. Furthermore, we would like to determine how sensitive these guarantees are to certain channel conditions and MAC parameterization, (e.g., channel utilization, average packet size, contention window sizes). We derive a closed-form formula for the delay of the packets originated from a single traffic flow on a channel occupied with background traffic. We use this analysis to guide the configuration of our modified DCF MAC.

Denote the mobile host sending the traffic flow under investigation as the “tagged host” and all other packets generated by other mobile hosts as the background traffic. Assume that each packet associated with background traffic has a transmission time L , which duration includes the time needed for RTS/CTS/ACK transmissions as well. Assume that the time between the last bit of a background packet and the first bit of the next background packet is exponentially distributed with average $1/\lambda$. Also assume the tagged traffic only occupies a small portion of the total channel utilization, (i.e., its effect on the background traffic is negligible).

The average channel utilization U can be approximated as

$$U = \frac{L}{L + \frac{1}{\lambda}}$$

$$\lambda = \frac{U}{L - L \cdot U}.$$

When a tagged packet arrives, the mobile host senses the channel and sends the packet if the channel appears to be idle. If the channel is busy or a collision occurs, the MAC invokes the backoff procedure and delays the transmission, otherwise, the tagged packet is sent. Assume each tagged packet has a transmission time m , and $m < L$. Denote d' as the average delay conditional on the backoff procedure. The average delay d of the tagged packet can be approximated as

$$d = U \cdot d' + (1 - U) \cdot m. \quad (1)$$

Denote d'_i as the total deferred time during the i th backoff period. According to the IEEE 802.11 protocol, the backoff timer is only decreased when the channel is idle. Denote b_i as the

random deferred time chosen by the DCF algorithm during the i th backoff, where b_i is a uniformly distributed random variable in the interval $[0, CW_i]$ times the length of a backoff time slot T_{slot} . During the i th backoff period, a number of background packets k_i are sent. Because the idle time between two background packets is exponentially distributed, k_i is a Poisson random variable with average λb_i . In the first backoff, the delay also includes the residual background packet length L' , which causes the backoff in the first place. In the subsequent backoffs caused by collisions, the delays include the length of the colliding background packet L .

Adding all the above together, the i th deferred time d'_i can be written as

$$d'_i = \begin{cases} L' + k_i L + b_i, & \text{for } i = 1 \\ k_i L + b_i + L, & \text{for } i > 1 \end{cases}.$$

The probability of collision after a backoff, denoted as p , can be estimated as the probability that a transmission attempt of a background packet starts exactly in the same time slot as chosen by the tagged host, otherwise the tagged station would sense the packet and could avoid collision

$$p \approx \lambda T_{\text{slot}}.$$

The average value of the total accumulated deferred time, $d' = E[\sum d'_i]$, takes into account occasional retransmissions and consecutive backoffs, and can be estimated as

$$d' = \sum_{i=1}^{\infty} E \left[\sum_{j=1}^i d'_j \mid i \text{ backoffs} \right] (1-p)p^{i-1} + m$$

$$= \sum_{i=2}^{\infty} \left(\sum_{j=2}^i E[(k_j + 1)L + b_j \mid i \text{ backoffs}] \right. \\ \left. + E[L' + k_1 L + b_1 \mid i \text{ backoffs}] \right) (1-p)p^{i-1}$$

$$+ E[L' + k_1 L + b_1 \mid 1 \text{ backoff}] (1-p) + m.$$

The contention window CW ranges from $2^{W_{\min}}$ to $2^{W_{\max}}$. In the j th backoff period, the backoff time b_j is chosen randomly in the range of $[0, 2^{W_{\min}+j-1}] T_{\text{slot}}$ until we reach the maximum backoff time, when it is chosen from $[0, 2^{W_{\max}}] T_{\text{slot}}$. The average backoff time is thus

$$E[b_j] = \begin{cases} T_{\text{slot}} \cdot 2^{W_{\min}+j-2}, & \text{for } 1 \leq j \leq W_{\max} - W_{\min} + 1 \\ T_{\text{slot}} \cdot 2^{W_{\max}-1}, & \text{for } j > W_{\max} - W_{\min} + 1 \end{cases}.$$

In the j th backoff period, the randomly chosen backoff time is b_j . Given this choice, the average number of background packets that arrive before the backoff timer expires is $E[k_j \mid b_j] = \lambda b_j$. The average number of packets is thus $E[k_j] = \lambda E[b_j]$. The average residual packet time is $E[L'] = L/2$.

Let $u = W_{\min} - 1$, $v = W_{\max} - W_{\min}$. Given this notation, the final closed-form result is

$$d' = 2^u \cdot T_{\text{slot}} \cdot (L\lambda + 1) \left(\frac{1 - (2p)^{v+1}}{1 - 2p} + 2^v \frac{p^{v+1}}{1 - p} \right) \\ + \frac{L}{1 - p} - \frac{L}{2} + m.$$

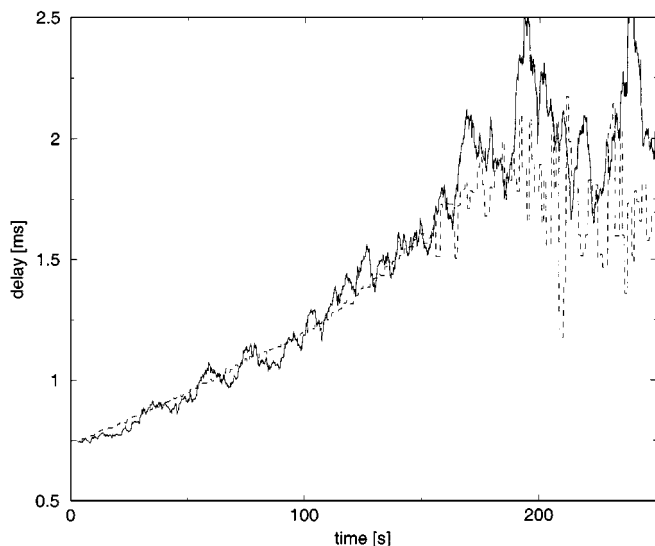


Fig. 1. The comparison of the analysis (shown as the dotted line) and measured results. Packet delays of single session versus increasing background load (running average) for a channel rate of 11 Mb/s.

Putting d' in (1), we have the estimated average delay d if we know the channel utilization U .

Fig. 1 shows a comparison between the analysis and the measured delay from our differentiated services wireless testbed. We compared the measured average delay of a tagged host with increasing levels of background traffic load. The tagged session generates 120-byte-long packets every 0.02 s, the length of background packets is 1500 bytes, and the channel rate is 11 Mb/s. The background traffic rate is gradually increased to the saturation point in small incremental steps. At every step, the average delay of tagged packets is calculated. The result shows that the estimated average delay closely matches the measured delay. Section V presents a detailed description of our wireless testbed and the configuration used for these results.

C. Discussion on Backoff Timers and Service Differentiation

We have previously described 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, and so the longer the time between transmission attempts.

Backoff times are set to a random value in the range $[0, CW] \cdot T_{\text{slot}}$. 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 (i.e., premium service) 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 the high-priority packets remain lower than that of low-priority packets, with the result of experiencing smaller average delays. Intuitively, even

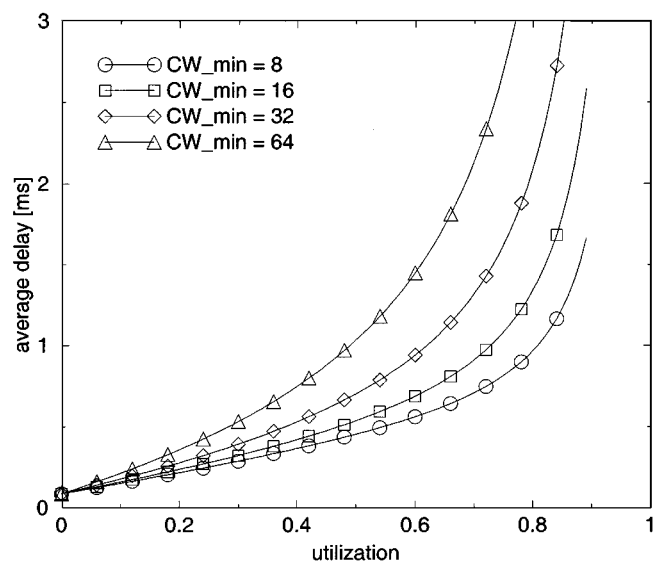


Fig. 2. Estimated average delays for different values of CW_{min} and increasing level of channel utilization, while $CW_{\text{max}} = 1024$ is kept constant. Channel rate is 11 Mb/s.

during highly congested periods, all classes have 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.

The analysis in Section III-B can be used to address the issue of how backoff values impact the average MAC delay for different levels of channel load. Only W_{min} and W_{max} values have to be modified in the equations accordingly. Fig. 2 shows the estimated average delays for increasing levels of utilization for several choices of $CWC_{\text{min}} = 2^{W_{\text{min}}} = 8, 16, 32,$ and 64 .

The analysis shows that, by setting different values of CW, differentiated levels of service can be achieved. We note, however, that the results of the analysis should be treated as qualitative results only since some of the assumptions made in the model are too simple when one considers highly bursty traffic scenarios. In Section III-D, we simulate realistic traffic mixes for TCP and UDP sources, and explore the achievable service differentiation using this simple means of control.

D. Evaluation of the Modified MAC to Support Service Differentiation Using Simulation

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. We use network simulation for the evaluation of the proposed mechanisms. For simulation, we use the ns-2 network simulator developed by the VINT Project [14] with the wireless extension produced by the MONARCH Group [23].

The traffic mix we consider consists of five mobile hosts sending high-priority voice traffic and ten mobile hosts starting

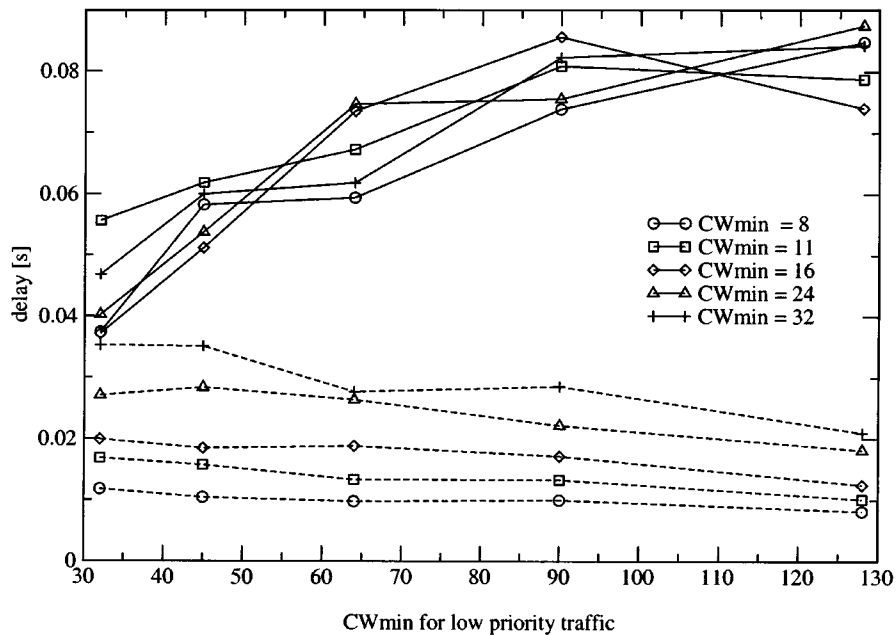


Fig. 3. Average delay experienced by voice and TCP flows for varying values of CW_{min} . The x axis is the CW_{min} for best-effort traffic. Different symbols represent different CW_{min} for high-priority traffic. Channel rate 2 Mb/s.

best-effort greedy TCP connections. Voice traffic was modeled using an on/off source with exponentially distributed on and off periods of 300 ms average each. Traffic was generated during the on periods at a rate of 32 kb/s with a packet size of 160 bytes, thus, the inter-packet time was 40 ms. During all simulations, the channel rate was 2 Mb/s.

We ran a set of tests for this traffic mix with varying values of CW_{min} for both traffic classes. For high-priority traffic, the CW_{min} values varied between (8,32), and the CW_{min} for best-effort traffic varied between (32,128). A value of 32 is the proposed by the standard [4], which applies to the case when only the best-effort traffic class is supported. We chose this value to be the delimiter between the two traffic classes under test. The values chosen for the high-priority traffic range is below this value. A value of 8 is proposed by the standard as an absolute minimum. By using the values above 32 for the 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 Section III-C, the maximum contention window for 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, five values were chosen to cover each range of CW_{min} values. Simulations were performed for all 5×5 combinations covering the whole plane. Packet delays were logged for both high- and low-priority traffic classes.

Fig. 3 shows the summary of the simulation results. The x axis corresponds to the CW_{min} of the best-effort packets. It can be observed that increasing this value results in larger delays for best-effort traffic and somewhat decreasing delays for real-time traffic. The delay for real-time sources is more significantly affected by their CW_{min} values (see the dashed lines), while the delays of best-effort packets are not affected greatly by the value chosen for real-time sources (straight lines). For 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 $CW_{min}^{lowprio}$.

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.

During simulation, the channel load is increased by adding a new voice, video (64 kb/s constant rate source) and TCP session periodically every 5 s. 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. 4 shows the delay throughout the simulation for the three traffic types. It can be observed that the delay increases for all service 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. 5 shows that the modified MAC enables the best-effort adaptive TCP traffic to utilize any free capacity unused 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 nondeterministic nature of service separation.

The modified MAC provides good service differentiation in terms of throughput and delay over a wide range of high-priority and best-effort traffic mixes. We investigate more dynamic traffic scenarios in Section VII.

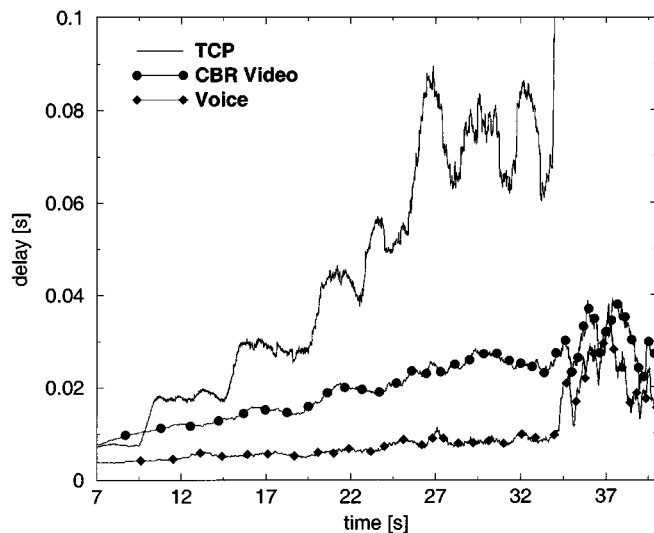


Fig. 4. Average delay experienced by gradually increasing the number of TCP and real-time sources over time. Channel rate 2 Mb/s.

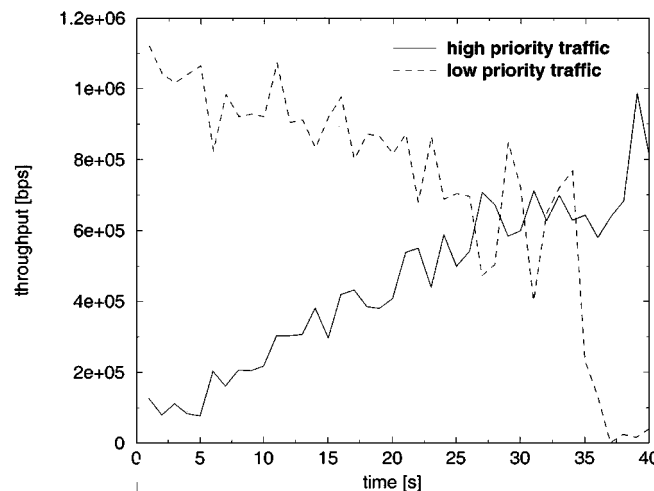


Fig. 5. Aggregate throughput of high-priority and best-effort traffic classes: number of TCP and real time sources increase over time. Channel rate is 2 Mb/s.

IV. ESTIMATION OF AVAILABLE RESOURCES USING A VIRTUAL MAC ALGORITHM

Many aspects of the wireless channel preclude exact control of resources (e.g., channel fading or interference). Furthermore, 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 Section III ensures effective service differentiation even in the case 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 traffic, as in most cases, applications require absolute and not relative service quality, (e.g., for voice or video). If a mobile host realizes that the channel is not able to meet its delay and loss requirements, it can either refrain from loading the channel or reduce application traffic demands, (e.g., by increasing compression). In order to make this decision, the host has to rely on accurate estimations of the achievable QoS of the radio channel.

The difficulty with this problem is that measuring simple channel properties, such as channel utilization, is not sufficient to estimate the loss and delay statistics of a new session. The reason for this is that the actual QoS depends on a number of factors, (e.g., the actual arrival pattern of packets or the ratio of hidden terminals). The analytic models published in the literature usually focus on one of these aspects and make a number of assumptions about the other aspects. Furthermore, the traffic models used are usually simplistic for real traffic scenarios, (e.g., assuming only long, greedy sessions). Even if the analytic models were more accurate and could take into account the relevant modeling issues, parameterizing them would be an extremely difficult task.

To overcome the problem of channel modeling, we take a more pragmatic approach: instead of modeling the interaction of MAC, the radio channel and background traffic load, we introduce a Virtual MAC (VMAC), which emulates the behavior of the MAC performance. We argue that the MAC algorithm is accurate, can be easily implemented, and scales to high data rates. To prove these claims, we implemented the VMAC algorithm on a mobile host accessing an 11-Mb/s wireless LAN. The efficiency of the algorithm and the implementation are discussed in Section IV-A.

A. Operation of the Virtual MAC Algorithm

The VMAC algorithm operates in parallel to the real MAC in the mobile host but the VMAC does not handle real packets; rather, it handles “virtual packets.” Scheduling these virtual packets on the radio channel is performed in the same way as real packets, which means channel testing and random backoff is performed as necessary. The difference arises when the VMAC decides to send a virtual packet. Unlike the case of real packets, no packet is transmitted. Rather, the VMAC algorithm estimates the probability of collision if the virtual packet were “really” sent. To make the algorithm conservative, a collision is “detected” whenever any other mobile station chooses the same slot for transmission (i.e., the channel is occupied by any station within the same slot time). In this case, the VMAC enters a backoff procedure, as a real MAC would do, after a collision had occurred.

For a real MAC, collision detection is realized using a timer which expires if neither a CTS in response to an RTS nor an ACK in reply to a data packet arrives in time, depending on the operation. If no CTS or ACK has been received before this timer expires then the real MAC assumes that a collision has occurred and the packet must be retransmitted. At this point, a real MAC would begin a backoff procedure. The VMAC does not detect collisions in this manner. Rather, it decides that a collision would have happened if a transmission occurs in the timeslot determined by its congestion avoidance algorithm. In other words, the VMAC 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.

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 $d = t_{\text{defer}} + t_{\text{RTS}} + t_{\text{CTS}} + t_{\text{packet}} + t_{\text{ACK}} + 3t_{\text{SIFS}} + 3\tau$ where τ is an estimate of

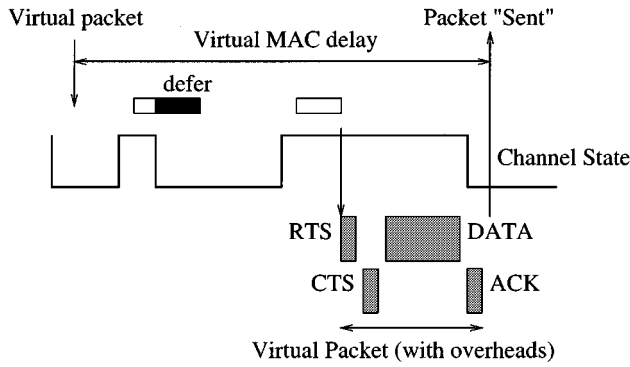


Fig. 6. An example of the operation of the VMAC algorithm. The channel state indicates an idle (state is high) or busy (low) channel. A virtual packet arrives during a busy period and the deferred timer is decremented during a short idle period, and virtual transmission happens during the next idle period, when the deferred timer expires.

the maximum propagation delay). An example of the operation of the VMAC is illustrated in Fig. 6.

The VMAC emulates not only backoff and collision resolution aspects of the real MAC but also all other aspects of a real MAC. For example, packet loss is signaled by the VMAC if the maximum number of “retries” is reached. It also adjusts the contention window by doubling the window until it reaches CW_{max} , when it stops increasing it further. If a transmission is successful then the contention window is reset.

After every successful or failed “transmission,” the VMAC waits for the next virtual packet to process. If, for example, the packets arrive at the VMAC at a rate of 20 ms and with size of 80 bytes, the output of the VMAC algorithm will closely match the delays experienced by a real constant rate encoded voice application.

One of the key advantages of the VMAC algorithm over analytic models is that it does not produce just a small set of performance measures, (i.e., estimates of first-order statistics). Rather, it produces a time series that can be identically analyzed to a time series produced by a real test. Consequently, there is no limit on using higher order statistics, which makes it possible to apply more sophisticated analyses and traffic control methods. For example, not only the n th moments of the delay can be calculated but also percentiles, burstiness, traffic envelopes, number of errored seconds, etc., which are more closely related to user perceived quality measures.

A disadvantage of the virtual algorithm is that it consumes extra processing capacity in the mobile host. This extra capacity is probably relatively small, since the VMAC does not have to operate continuously, it *does not* transmit packets, and computationally its complexity is equivalent to running the original MAC algorithm.

The VMAC can be applied to estimate the performance of either best-effort or better than best-effort traffic by changing the MAC mechanism to match the changes discussed for service differentiation in Section III. These estimates can be used for a variety of traffic control algorithms. In the proposed architecture, we use the VMAC algorithm to estimate the QoS of better than best-effort traffic and base the admission decision on that estimate.

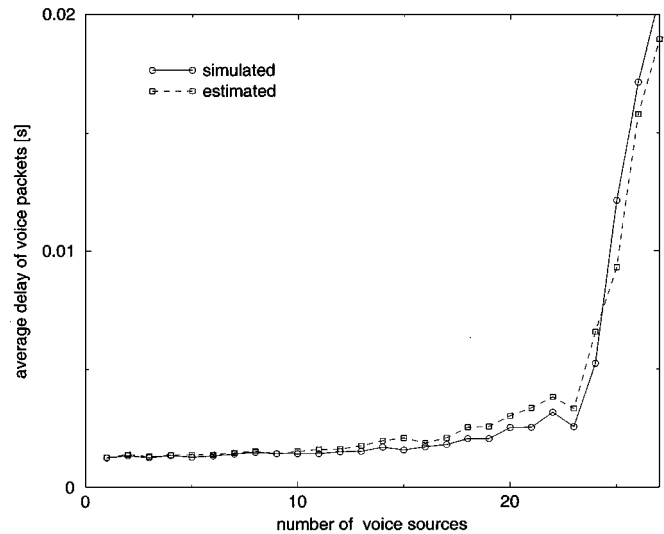


Fig. 7. Virtual and simulated average MAC delay of a new voice source versus the number of active voice sources.

B. Evaluation of the VMAC Algorithm

Fig. 7 shows results from a simulation test of the efficiency of the VMAC 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 over 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. 8 shows the results for a more complex simulation test where 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 10 kbytes 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 10 s. This creates a highly bursty background data traffic load with multiple time-scale fluctuations [22], [24], [25]. The TCP load is sufficient to saturate the channel by itself.

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 Section IV-A, for the voice source. The results show that the delay is efficiently estimated by the VMAC 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 burstier, does not only increase the voice packets’ delay but also increases the delay variance, as shown in Fig. 8.

V. IMPLEMENTATION OF THE VMAC ALGORITHM IN A WIRELESS TESTBED

Since the wireless DiffServ MAC can only offer soft and relative differentiation, it is important that the mobile hosts can

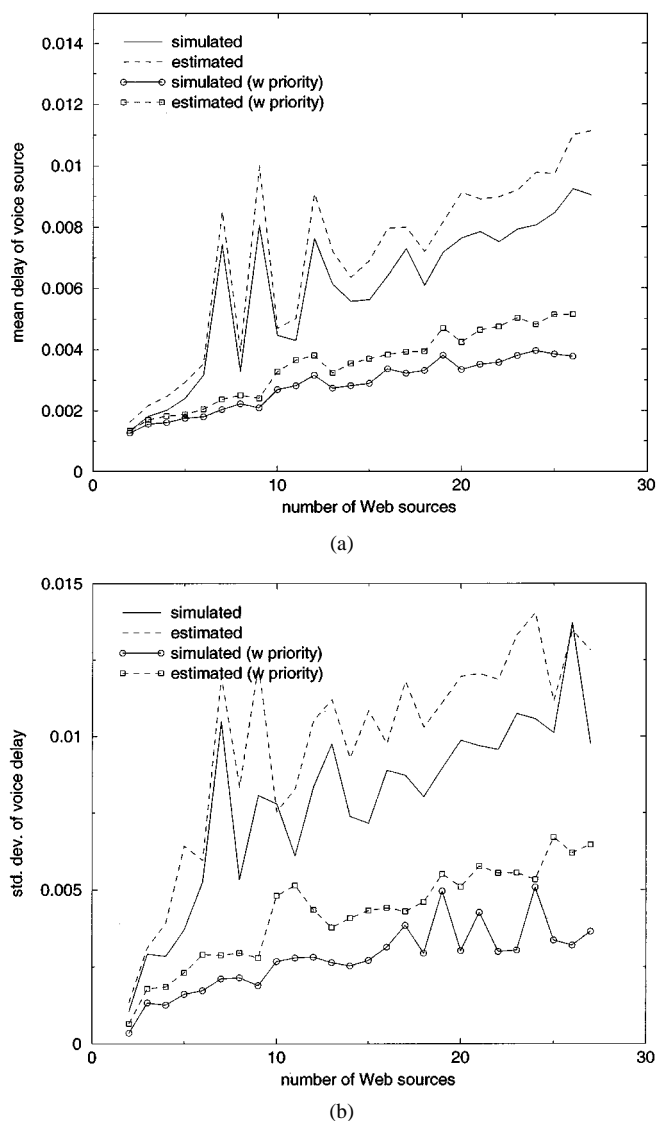


Fig. 8. Average delay (top) and delay variation (bottom) of a new voice source obtained by simulation and from the VMAC algorithm, versus number of Web sources with and without priority for voice traffic. Channel rate is 2 Mb/s.

accurately estimate the channel. There are several issues concerning the VMAC that can only be satisfactorily evaluated in a real wireless network with real applications. In what follows, we describe a wireless differentiated services testbed and its VMAC implementation. In addition, we compare the estimates given by the VMAC and the performance perceived by real applications.

The VMAC was implemented on a Linux machine with a modification to the wireless card's device driver. We used 11-Mb/s Lucent and Aironet PCMCIA cards in the experiments. These cards, with the modified drivers, are capable of capturing all "overheard" layer-two transmissions, (e.g., CTS, RTS, ACK packets, even with CRC errors). Packets were timestamped with approximately microsecond precision. This traffic trace was used as input to the VMAC algorithm. In a commercial implementation, the VMAC could be placed into the firmware of the wireless card and would operate in real-time.

The testbed generates traffic mixes of TCP and UDP flows, with different levels of offered load, as illustrated in Fig. 9.

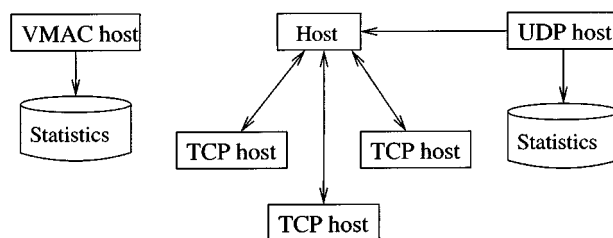


Fig. 9. Testbed configuration.

The wireless testbed consists of six hosts with 11-Mb/s IEEE 802.11 PCMCIA cards. All mobile hosts were configured to operate in DCF *ad hoc* mode. Three of the mobile hosts (indicated as TCP hosts) were used to generate random TCP traffic. The hosts transferred random length files independently of each other using TCP. The average file size was 50 kbytes. Between file transfers, each host waited a random duration before the next transfer was started. The load on the channel was modified by adjusting the average idle time. The UDP host generates packets every 20 ms at a data rate of 32 kb/s, resulting in a voice-like traffic stream.

Because the cards do not support APIs to change the contention window limits, all sources use the same backoff algorithm, using factory-set default values. Therefore, we were not able to evaluate the previously proposed DiffServ MAC but we could still evaluate the accuracy of the VMAC algorithm.

The UDP host logged the delays of the wireless MAC. This was achieved by modifying the wireless card network driver to capture all packet processing events together with an accurate timestamp at a resolution of approximately 1 μ s. The resulting log file consists of packet arrivals to the MAC, packet sizes, MAC deferred delays, and indications of successful or unsuccessful delivery.

The fifth host acted as a traffic monitor and executed the VMAC algorithm (indicated in the figure as the VMAC host). The VMAC host logged a similar file as the UDP host but this file consisted of estimated delays provided by the VMAC algorithm.

During the experiment, the channel utilization gradually increased up to its saturation point by decreasing the average idle time from 10 to 0 s. Fig. 10(a) shows the physical level channel utilization versus time. The maximum channel utilization reached was approximately 70%.

The measured UDP delay statistics and the estimated delay statistics from the VMAC algorithm are shown in Fig. 10(b). It can be observed that the VMAC implementation could estimate the measured delay with excellent precision during the entire experiment for all channel loads. Thus, mobile hosts running passive monitors and VMAC's can rely on precise quality feedback for traffic control purposes.

VI. ESTIMATION OF APPLICATION-LEVEL QOS USING A VIRTUAL SOURCE ALGORITHM

The VMAC measures virtual packet delays, packet losses, and collisions at the MAC level. The delay experienced by an application can be very different than the delay provided by

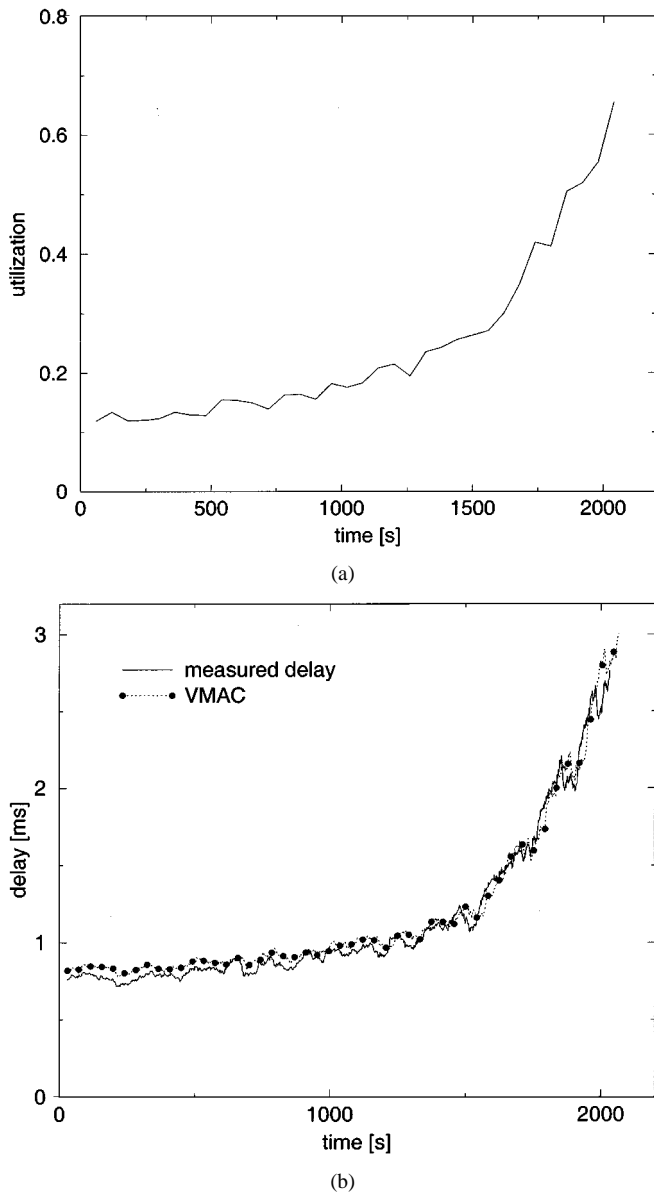


Fig. 10. (a) Measured channel utilization with increasing TCP traffic. (b) Average delay of UDP traffic. The channel rate is 11 Mb/s.

the MAC. The reason for this is that application-level data usually has to be packetized, encoded, and placed into an interface queue before it is received by the MAC layer.

Also, even the estimated MAC layer delay depends not only on the channel but also the arrival pattern of packets at the MAC. This effect can be due to the correlation structure of the traffic load on the channel. These factors are taken into consideration by the VS algorithm. For certain applications, running the VS can provide more precise estimates of the achievable performance. In addition, the VS makes it possible to tune certain application level QoS parameters.

The VS algorithm consists of a Virtual Application, interface queue, and VMAC. 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 in a virtual buffer. When the virtual packet is finished

processing in the VMAC, the total delay is calculated comparing the actual time to the timestamp stored in the packet.

Although the VS gives a more useful estimation for an application, the VS is not as generally applicable as the VMAC, since it requires that the application traffic is well known in advance. Nevertheless, we believe that there are a number of important applications that fit into this category such as constant bit-rate encoded voice or video. If the application traffic is not easy to emulate, then traffic management can fall back to the estimation provided by the VMAC fed with a generic traffic process (e.g., packets with constant or exponentially distributed interarrival times).

A. Virtual Delay Curves

The application delay depends on several factors. Certain factors depend on the application (e.g., packet size, packet rate), while others depend on the load of the channel. The VS algorithm monitors the channel continuously and 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 bit rate, 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 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 bit rate, there is a tradeoff between packetization delay and MAC delay.

Denote the function $d(p_{\text{rate}})$ as the virtual delay curve of an application, where p_{rate} is the packet inter-arrival time of the application, e.g., $p_{\text{rate}} = 0.02$ packets per second for voice, but the data bit rate is kept constant, i.e., $p_{\text{size}} \cdot p_{\text{rate}} = \text{const}$ (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. Delay curve can be constructed from the virtual packet delays obtained from the VS algorithm. Similarly, we can define the virtual delay variance curve $v(p_{\text{rate}})$ which calculates the virtual delay variances, respectively. Based on the delay curve, a mobile host or base station can choose the optimal packet rate and packet size so that an application experiences minimum delay and delay variance.

Fig. 11 shows the virtual delay and variance curves for 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 send 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.

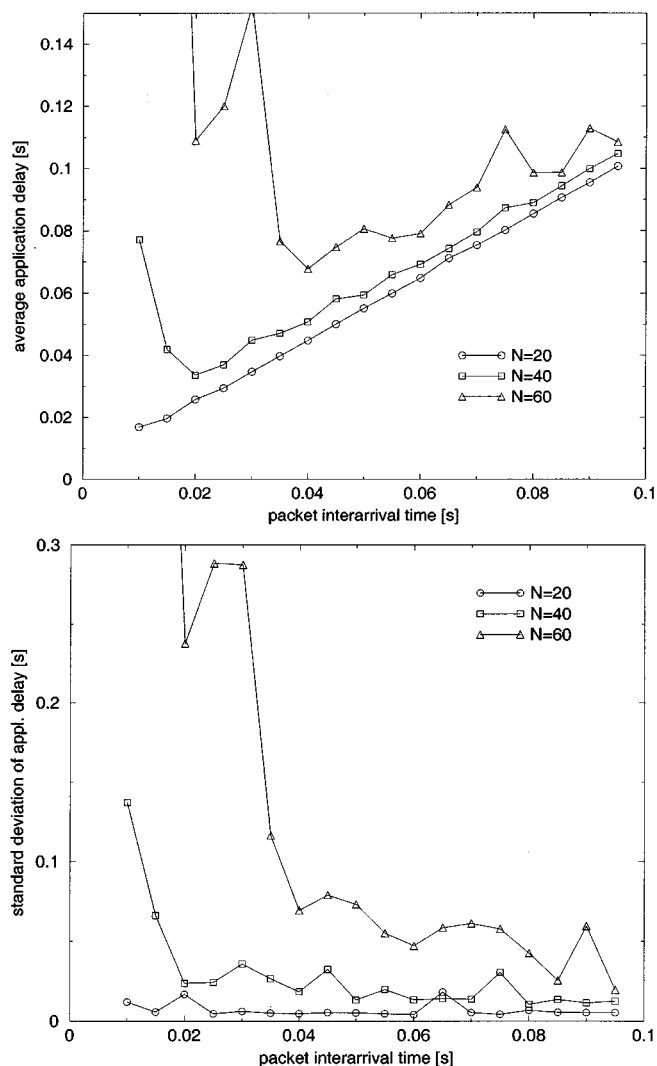


Fig. 11. Virtual delay and delay variance curves at several radio channel loads, N denotes the number of background Web sessions. Web traffic uses $CW_{min} = 31$ and the virtual algorithm uses $CW_{min} = 8$. The channel rate is 2 Mb/s.

VII. DISTRIBUTED ADMISSION CONTROL ALGORITHM IN A MULTICELL ENVIRONMENT

The estimates provided by the VMAC and the VS can be used by a mobile host before actually starting transmission. Because the virtual algorithms do not require high processing capacity and do not load the channel, they may run continuously and not only when a service request arrives. In other words, the virtual algorithms are designed to continuously keep track of the health of the channel.

This estimate can be used to apply traffic control to maintain the congestion of the channel at a low level and the relative performance guarantees provided by the DiffServ MAC at absolute levels. There are numerous ways to utilize these estimates from the VMAC and VS algorithms. For example, elastic, best-effort traffic can be policed or shaped in response to estimation of congestion. Premium, delay-sensitive sessions are usually not elastic, thus admission control is more appropriate to control them. In this section, we apply the latter type of traffic control. However, we note that adding some sort of control for best-effort traffic may further improve the quality assurances. Every

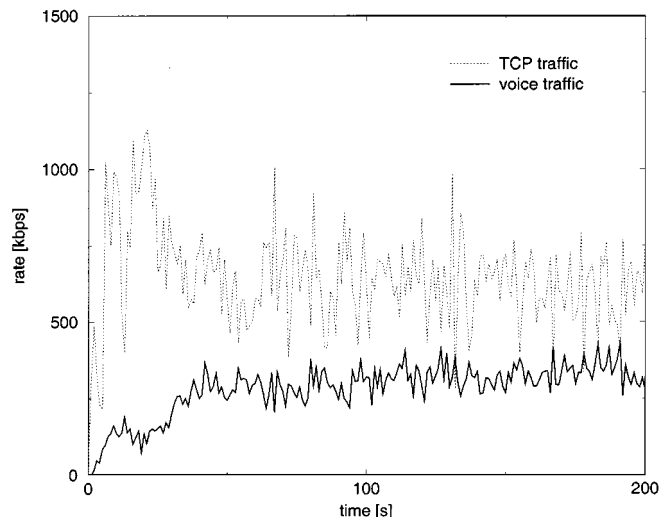


Fig. 12. Aggregate rates of TCP and voice traffic in the entire service area.

mobile host keeps track of the state of the channel using either VMAC or VS. The admission control algorithm compares the results of the VS and VMAC with the service requirements and admits or rejects a new session accordingly. For admission, we only use the average delay estimation over the last five seconds. The admission algorithm runs in every mobile host and is performed in a fully distributed and autonomous manner.

Because the radio channel properties may be different at the receiving and transmitting mobile hosts, it is preferable that both hosts execute the VS and VMAC algorithms to ensure that the service quality will be met for a new session. This can be executed during session setup. Admission is granted if both virtual algorithms at the mobile hosts admit the new request.

In this section, we investigate this concept through simulation of a complex configuration with random topology and random traffic. The aim is to test how the modified MAC and VMAC algorithm perform in the presence of a highly dynamic real-time and non real-time traffic mix when the radio channel is dynamically shared among traffic streams between mobile hosts and base stations.

Ten base stations were placed randomly on a 400 m by 400 m rectangular area with their coverage areas significantly overlapping. One hundred mobile hosts were placed randomly in the coverage area. Every mobile host was associated with the nearest base station. Half of the mobile hosts randomly generate Web sessions and the other half randomly generated voice traffic. The length of the voice sessions and the inter-arrival times between connection requests were exponentially distributed. The average session length was 30 s. Upon completion of a session, a mobile host attempted a new call after an average waiting period of 10 s.

Independent VS algorithms running in all base stations continuously monitored the radio channel. Admission control was applied to delay sensitive voice sessions. When the estimated delay exceeds 10 ms, new voice sessions were 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. There was no admission control applied to Web traffic.

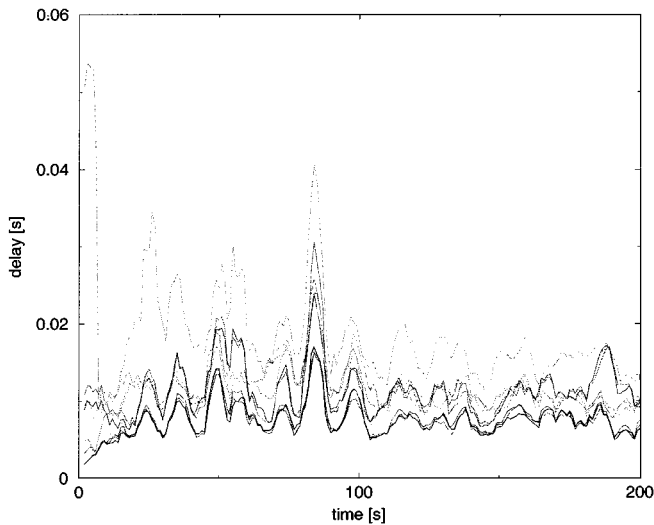


Fig. 13. Estimated delays by VS algorithms running at base stations.

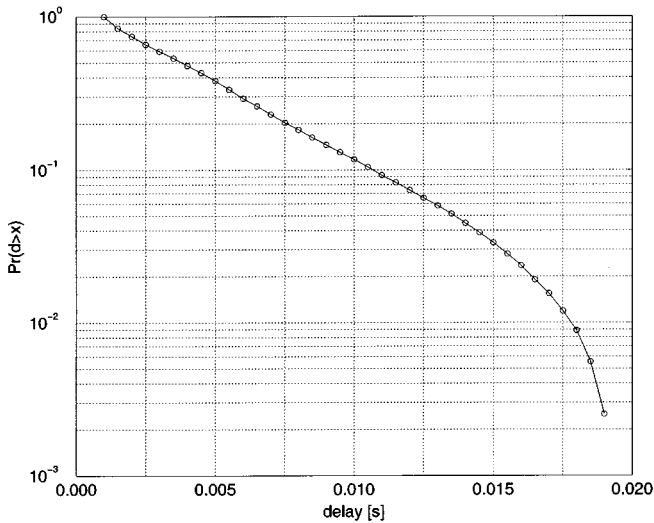


Fig. 14. Delay distribution of voice packets.

Fig. 12 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. 13 shows the delay estimations by the VS algorithms running in base stations. It can be observed that the delay estimation is kept 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 did not accept voice traffic during these periods. On the other hand, other base stations were continuously in the accept state. This was due to the overlapping of cells and the shared radio channel.

Fig. 14 shows the empirical distribution of voice packet delays from accepted sessions. The low delays experienced indicate that the overall channel state is efficiently controlled by the distributed monitorings and admission control algorithm, even in the presence of highly dynamic TCP traffic.

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. We evaluated the efficiency of the VMAC algorithm using simulation and implementation in an experimental differentiated services wireless testbed. 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.

Finally, we are currently building a wireless DiffServ testbed using Cellular IP [13] 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.

ACKNOWLEDGMENT

The authors would like to thank G.-S. Ahn and J.-K. Oh for their valuable contribution and help. M. Barry and A. 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. Finally, the authors would like to thank Prof. M. Schwartz for his input.

REFERENCES

- [1] M. Mouly and M. B. Pautet, "The GSM System for Mobile Communications," Telecom Pub, 1992.
- [2] "Digital Cellular Telecommunications System General Packet Radio Services: Service Description EN 301 344 v6.3.2," ETSI, 1999.
- [3] T. Nandagopal, S. Lu, and V. Bharghavan, "A unified architecture for the design and evaluation of wireless fair Queuing algorithms," in Proceedings of MOBICOM, Seattle, WA, Aug. 1999.
- [4] *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," in *IEEE Performance and Dependability Symp.*, Raleigh, NC, Aug. 1998.
- [6] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A media access protocol for wireless LAN's," in *Proc. ACM SIGCOMM*, London, UK, Sept. 1994.
- [7] I. F. Akyldiz, J. McNair, L. C. Martorell, R. Puigjaner, and Y. Yesha, "Medium access control protocols for multimedia traffic in wireless networks," *IEEE Network*, vol. 13, pp. 39–47, July 1999.
- [8] C. C. Chow and V. C. M. Leung, "Performance of IEEE 802.11 medium access control protocol over a wireless local area network with distributed radio bridges," in *Proc. WCNC*, New Orleans, LA, Sep. 1999.

- [9] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "RFC 2475 An Architecture for Differentiated Services," Internet Engineering Task Force, 1998.
- [10] S. Lu, V. Bharghavan, and R. Srikant, "Fair scheduling in wireless packet networks," in *Proc. ACM SIGCOMM*, Cannes, France, 1997.
- [11] V. Paxson and S. Floyd, "Wide area traffic: The failure of poisson modeling," *IEEE/ACM Trans. Networking*, vol. 3, pp. 226–244, June 1995.
- [12] M. A. Visser and M. E. Zarki, "Voice and data transmission over an 802.11 wireless network," in *Proceedings of PIMRC*, Toronto, Canada, Sept. 1995.
- [13] A. Valko, "Cellular IP—A new approach to internet host mobility," *Comput. Commun. Rev.*, vol. 29, pp. 50–65, Jan. 1999.
- [14] L. Breslau and D. Estrin *et al.*, "Advances in network simulation," *IEEE Computer Mag.*, vol. 33, pp. 59–67, May 2000.
- [15] T. Nandagopal, T.-E. Kim, X. Gao, and V. Bharghavan, "Achieving MAC layer fairness in wireless packet networks," in *Proc. MOBICOM*, Boston, MA, Aug. 2000.
- [16] N. H. Vaidya, P. Bahl, and S. Gupta, "Distributed fair scheduling in a wireless LAN," in *Proc. MOBICOM*, Boston, MA, Aug. 2000.
- [17] F. Cali, M. Conti, and E. Gregori, "IEEE 802.11 wireless LAN: Capacity analysis and protocol enhancement," *Proc. IEEE INFOCOM*, Apr. 1998.
- [18] G. Bianchi, L. Fratta, and M. Oliveri, "Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LAN's," in *Proc. PIMRC*, Taipei, Taiwan, Oct. 1996.
- [19] J. Weinmiller, M. Schlager, A. Festag, and 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.
- [20] J. Weinmiller, H. Woesner, J. P. Ebert, and A. Wolisz, "Analyzing and improving the 802.11-MAC protocol for wireless LAN's," in *Proc. MASCOT*, San Jose, CA, Feb. 1995.
- [21] J. L. Sobrinho and A. S. Krishnakumar, "Real-Time traffic over the IEEE 802.11 medium access control layer," *Bell Labs Tech. J.*, Autumn 1996.
- [22] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic," in *ACM SIGComm '93*, San Francisco, CA, Sept. 1993.
- [23] CMU MONARCH Project. [Online]. Available: www.monarch.cs.cmu.edu
- [24] M. S. Taqqu, W. Willinger, and R. Sherman, "Proof of a fundamental result in self-similar traffic modeling," *Comput. Commun. Rev.*, vol. 27, pp. 5–23, 1997.
- [25] M. E. Crovella and A. Bestavros, "Self-Similarity in world wide web traffic: Evidence and possible causes," in *Proc. ACM SIGMETRICS*, May 1996, pp. 160–169.
- [26] T. Nandagopal, T. Kim, P. Sinha, and V. Bharghavan, "Service differentiation through end-to-end rate control in low bandwidth wireless packet networks," in *IEEE Int. Workshop on Mobile Multimedia Communications*, San Diego, CA, Nov. 1999.



Andrew T. Campbell received the Ph.D. degree in computer science from Lancaster University, U.K., in 1996.

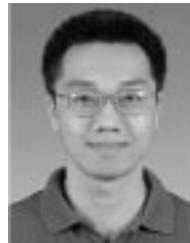
He is an Associate Professor in the Department of Electrical Engineering and member of the COMET Group, Columbia University, New York, NY. His areas of interest encompass mobile computing and networking, open programmable networks, and quality-of-service research. He is active in the IETF and the international working group on Open Signaling (OPENSIG).

Dr. Campbell received the NSF CAREER Award for his research in programmable mobile networking in 1999.



Michael Barry (M'90) received the B.S. and M.S. degrees in computer engineering from the University of Limerick, Limerick, U.K., in 1995 and 1999, respectively.

Since 1998, he has been working in Broadcom Eireann Reserach Limited. In 2000, he was a visiting Scholar with the COMET group at the University of Columbia, New York, NY. His research interests include Mobile IP, IP QoS, and wireless access mechanisms.



Li-Hsiang Sun (S'00) received the M.S. degree in electrical engineering from Columbia University, New York, NY, in 1999. He is currently working toward the Ph.D. degree in the Electrical Engineering Department of the same university.



Andras Veres received the M.Sc. degree in electrical engineering in 1996. He is currently pursuing the Ph.D. degree at the Department of Telecommunications and Telematics, Budapest University of Technology and Economics.

He is a Scientist with Traffic Laboratories, Ericsson Research, Hungary. His research interests include performance analysis of TCP/IP networks and wireless data networks.