AN EFFICIENT MULTIACCESS PROTOCOL FOR WIRELESS NETWORKS

Benjamin W. Wah and Xiao Su

Department of Electrical and Computer Engineering
and the Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
E-mail: {wah, xiao-su}@manip.crhc.uiuc.edu
URL: http://manip.crhc.uiuc.edu

ABSTRACT

In this paper, we propose and evaluate an efficient multiaccess protocol for cell-based wireless networks. Our protocol addresses the problems in existing random-access protocols for wireless networks: long-term fairness as well as short-term fairness in accessing a shared channel and the detection of hidden and exposed collisions. Our proposed protocol is a limited contention protocol in which the set of contending mobiles are chosen based on a global common contention window maintained by every mobile station. The contention window is adjusted based on three possible channel states: no transmission, success, and collision. We assume that the channel state at the end of each contention slot is broadcast by a base station in a control channel. We show analytically that the time interval between two successive accesses to the channel by any station is geometrically distributed, and that each station has equal chance to access the channel in every contention period. This is significantly better than existing random-access protocols based on the binary exponential backoff algorithm, which results in large variances in inter-access delays. Our experimental results also show that the number of contention slots to resolve collisions is constant on the average, independent of the number of contending stations.

1. INTRODUCTION

The design of an efficient and scalable medium access control (MAC) protocol is extremely important for wireless networks, where bandwidth is a precious and scarce resource. Existing work on wireless medium access control protocols can be classified into two categories: *ordered-access* and *random-access*. Ordered-access protocols, such as token-based and polling schemes, rely on knowledge of the network configuration in order to predetermine the use of a shared channel. They are usually very efficient when the network configuration is static, requiring constant overhead to resolve the transmission order. However, they do not work well in mobile networks in which stations can join and leave dynamically. For this reason, we study random-access schemes in this paper.

One of the popular random-access schemes used in mobile networks today is DFWMAC, a CSMA/CA protocol selected as the IEEE 802.11 draft standard [4]. Collisions in this protocol are resolved by a binary exponential backoff algorithm, similar to that used in Ethernets. There are two problems associated with the use of the backoff algorithm. First, although the algorithm is fair in the long term so that every station has equal access on the average, it is not fair in the short term because it does not give equal access to all the stations competing for the channel. Oftentimes, a station that has just transmitted has a higher chance to access the channel again in the near future. This behavior may cause large variations in inter-channel access delays, an undesirable phenomenon in systems wishing to provide certain qualify of service in access. Second, the protocol does not operate efficiently in the presence of hidden and exposed terminals [8]. The backoff counters are updated incorrectly for stations involved, and do not reflect the local contention level.

Our proposed wireless window protocol (WWP) is a limited contention protocol in which the set of contending mobiles are chosen based on a global common contention window maintained by every mobile. The contention window is adjusted based on three possible channel states: no transmission, success, and collision. We assume that the channel state at the end of each contention slot is broadcast by the base station in the downlink. Initially, each station generates a random contention parameter between zero and one based on a uniform distribution. Each station then derives a window with the goal of isolating exactly one parameter in the window. Since all stations derive the window boundaries using identical information and the same algorithm, the windows at all stations are synchronized. Depending on the state of contention (collision, idle, success) broadcast by the base station, stations update their windows in a synchronized fashion. Eventually, only one station is isolated in the window and transmits the message to the base station, which may forward it to another mobile in the same cell.

Our protocol addresses the two problems associated with DFWMAC. Our analytical and experimental results demonstrate WWP's channel efficiency as well as its long-term and

Research supported by National Aeronautics and Space Administration Grant NAG 1-613 and by National Science Foundation Grant MIP 96-32316.

Proceedings of 1998 International Symposium on Internet Technology, Taipei, Taiwan, April 1998

short-term fairness. Further, as a base station always broadcasts reliable channel-state information to mobiles in the same cell, false interpretations of channel states in the hidden- and exposed-terminal scenarios are avoided in one cell. There are some implications in two-cell scenarios that are discussed in Section 3.

WWP bears certain similarity to binary-tree splitting protocols proposed in wired domains in its contention-resolution process. According to the tree splitting protocol, when a collision involving n stations happens, the stations are randomly split into two subsets by flipping a coin. The stations in the first subset retransmit in the next slot, whereas the second subset must wait until all the stations in the first subset have succeeded. If the *first transmission rule*, *i.e.*, when packets are transmitted for the first time, is incorporated, there are a few variants of the basic protocol. The most celebrated one, the *epoch mechanism*, was suggested by Gallager [6] and by Tsybakov and Mikhailov [1]. It achieves a maximum stable throughput of 0.4871.

The major difference between WWP and the epoch algorithm is that WWP is not a contention resolution algorithm in a strict sense. The objective of WWP is to fulfill one successful transmission in the least possible number of slots, whereas resolution algorithms resolve a whole set of stations that are involved in a collision before accepting new stations. Intuitively, contention-resolution algorithms may achieve higher channel efficiency, because they utilize information obtained from previous contentions. However, new stations suffer from longer delays. Our protocol achieves a balance between the channel throughput and the lag between the time when new stations join and the time when they are served.

There are two major advantages of WWP over the epoch algorithm. First, WWP does not put a stringent synchronization requirement on its implementation as the epoch algorithm. In the epoch algorithm, synchronization must be supported at least to the granularity of one tenth of a slot if one successful transmission requires four to five splits of the initial epoch. In WWP, synchronization is only required in the contention-slot boundary. Second, WWP does not adopt the Poisson arrival model as assumed by the epoch algorithm. As is well known, packet arrivals to the network cannot be modeled as a Poisson process since packets are bursty within connections, and the major part of the Internet traffic, such as Web surfing and ftp, is connection-oriented.

The rest of the paper is organized as follows. Section 2 presents WWP in a one-cell scenario. Section 3 describes modifications to WWP in order to adapt it to cell overlays in a two-cell scenario. It also discusses the differences between WWP and its Ethernet counterpart. Section 4 presents the performance evaluations of WWP and compares it to DFW-MAC in both the one-cell and the two-cell scenarios. Finally, Section 5 summarizes our work and discusses future plans.

2. WINDOW-BASED WIRELESS WINDOW PROTOCOL FOR ONE CELL

In this section, we present the design of WWP for a one-cell case. Section 2.1 gives an overview of the protocol. Since the key aspect of the protocol is the adjustment of windows based on the channel state and the current channel load, Section 2.2 discusses the dynamic-programming formulation of window adjustments. Section 2.3 presents WWP with lookahead technique. Finally Section 2.4 gives our analytical result on the inter-channel access delay.

2.1. Overview

In this section, we describe the operation of our proposed window-based protocol. The protocol can be described in a two dimensional space as illustrated in Figure 1. The time space shows the progression of contention slots, and the parameter space defines stations that are eligible to contend. The operation of the protocol in one contention period consists of the following steps.

1. *Parameter initialization.* A station ready for transmission generates a random *contention parameter* in the parameter space. Without loss of generality, we assume that the parameters are generated from a uniform distribution between 0 and 1. New stations arriving before the beginning of a contention period must wait until the beginning of the next contention period. Since stations regenerate their contention parameters every time in the beginning of a contention period, each station has an equal chance of accessing the channel in each period. (This is different from ordered-access schemes that schedule accesses after generating the contention parameters once.)

2. *Window estimation based on channel load.* Each station maintains a lower bound $\mathcal{L}$ and an upper bound $\mathcal{U}$ in the parameter space. (The bounds identify stations that can participate in the contention process.) Initially, $\mathcal{L} = 0$ and $\mathcal{U} = 1$. In addition, each station computes $\mathcal{W}$, $\mathcal{L} \leq \mathcal{W} \leq \mathcal{U}$, based on an estimated channel load. As each ready station starts with identical information and the same algorithm, $\mathcal{L}$, $\mathcal{U}$ and $\mathcal{W}$ in all stations are synchronized.

3. *Contention phase.* A station transmits a short control packet in the uplink if its contention parameter is between $\mathcal{L}$ and $\mathcal{W}$. It keeps quiet if its contention parameter is between $\mathcal{W}$ and $\mathcal{U}$. It drops out from the current contention period if its parameter is outside the range between $\mathcal{L}$ and $\mathcal{U}$.

4. *Broadcast of contention information by the base station.* All the stations whose contention parameters are in the range between $\mathcal{L}$ and $\mathcal{U}$ listen to the broadcast by the base station in the downlink in the second half of the contention slot.

5. *Window refinement phase.* If the base station indicates in its broadcast that the transmission in the first half of

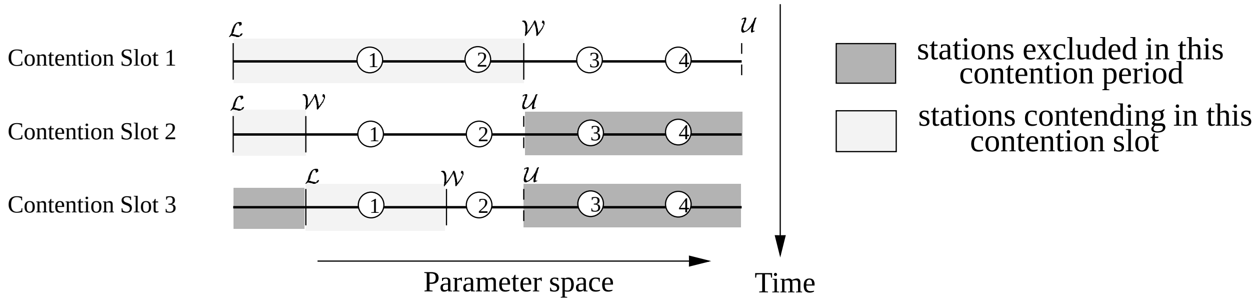


Figure 1: Window adjustments in one contention period. The station identifiers of the contention parameters are indicated by the circled numbers.

the slot was successful, then go to Step 6. If the base station indicates collision, then all mobile stations update $\mathcal{U}$ to $\mathcal{W}$. Finally, if the base station indicates an idle channel in its broadcast, then all mobile stations update $\mathcal{L}$ to $\mathcal{W}$. All stations whose parameters are between $\mathcal{L}$ and $\mathcal{U}$ compute a new value of $\mathcal{W}$ between $\mathcal{L}$ and $\mathcal{U}$ using dynamic programming (or from a lookup table computed ahead of time). Note that $\mathcal{L}, \mathcal{U}$ and $\mathcal{W}$ are synchronized in all participating stations without any additional broadcasts as they receive identical information and apply the same algorithm. Go to Step 3.

6. *Success.* The station that has just transmitted successfully can continue transmission to the base station. The message, if directed to another mobile station in the same cell, will be forwarded in the downlink. When done, the base station informs the remaining stations in the cell, and the contention period starts anew. Go to Step 1.

Figure 1 illustrates how windows are adjusted in one contention period. In the first contention slot, each of the four active stations generates a random contention parameter in $(\mathcal{L}, \mathcal{U}]$, and sets the initial window to $(\mathcal{L}, \mathcal{W}]$. Stations 1 and 2 have contention parameters in $(\mathcal{L}, \mathcal{W}]$ and proceed with transmission, whereas Stations 3 and 4 keep quiet since their contention parameters are out of the window. As the two stations transmit simultaneously, collision is detected by the base station and is broadcast in the downlink in the second half of the contention slot. In Contention Slot 2, all stations update $\mathcal{U}$ to $\mathcal{W}$ to record the new upper bound of the parameter space, and reduce $\mathcal{W}$ to some common value in $(\mathcal{L}, \mathcal{U})$. Since Stations 1 and 2 are outside $(\mathcal{L}, \mathcal{W})$, the base stations senses it to be idle and broadcasts the state in its downlink. In Contention Slot 3, Stations 1 and 2 update the contention window to the other half by setting $\mathcal{L}$ to $\mathcal{W}$, and compute a new $\mathcal{W} \in (\mathcal{L}, \mathcal{U}]$. Only Station 1 transmits in this slot, so successful transmission is detected.

2.2. Window Adjustments by Dynamic Programming

The efficiency of our proposed window protocol depends on the way that $\mathcal{W}$ is set in each contention slot. We formulate the choice of $\mathcal{W}$ as a dynamic programming optimization problem, with an objective of minimizing the future number of contention slots.

Let n be the number of initial contending stations for the contention period. (New arriving stations can only join at the beginning of a contention period.) Define the following notations, assuming $a < w < b$.

- $N(a, b)$: Minimum expected number of future slots to resolve contention, given that a collision occurs in the current window $(a, b]$.
- $P_{succ}(a, w, b, n)$: Probability of success in the next slot if window $(a, w]$ is used.
- $P_{col}(a, w, b, n)$: Probability of collision in the next slot if window $(a, w]$ is used,
- $P_{idle}(a, w, b, n)$: Probability of channel being idle in the next slot if window $(a, w]$ is used.

$N(a, b)$ can be derived recursively from the following formula:

$$N[a, b] = \min_{a<w<b} \{1 + 0 \times P_{succ}(a, w, b, n) + N[a, w] \times P_{col}(a, w, b, n) + N[w, b] \times P_{idle}(a, w, b, n)\} \quad (1)$$

Let $F(.)$ to be the cumulative distribution function (CDF) of contention parameters. The unknown probabilities are computed as follows.

$$P_{col}(a, w, b, n) = \frac{(1-F(a))^n - (1-F(w))^n - n(F(w)-F(a))(1-F(w))^{n-1}}{(1-F(a))^n - (1-F(b))^n - n(F(b)-F(a))(1-F(b))^{n-1}} \quad (2)$$

$$P_{idle}(a, w, b, n) = \frac{(1-F(w))^n - (1-F(b))^n - n(F(b)-F(w))(1-F(b))^{n-1}}{(1-F(a))^n - (1-F(b))^n - n(F(b)-F(a))(1-F(b))^{n-1}} \quad (3)$$

Details on how to arrive at the above equations can be found elsewhere [2, 7]. It follows that once the channel state and contention-parameter CDFs are known, an optimal window can be calculated. It can be shown that the CDF of contention parameters has little effect on the protocol's performance, provided that contention parameters are real numbers and the probability of two contention parameters having the same value is zero. Hence, without loss of generality, we assume that contention parameters are uniformly distributed in $(0, 1)$ in the rest of the paper. (When stations generate the same contention parameters, we assume that the stations regenerate another set of parameters when the window size is smaller than a prescribed threshold.)

To allow WWP to work efficiently and to compute the optimal $\mathcal{W}$, we need to know n, the number of contending stations. Since n is difficult to find exactly, we compute a maximum likelihood estimate of n based on the window bounds that have isolated the smallest contention parameter belonging to the winning station in the last contention period. The formulas can be found in [2, 7].

2.3. Lookahead WWP

When the uplink and downlink in WWP are implemented by different channels, then the base and the mobile stations can transmit simultaneously using different frequency bands. Since the result of contention in one slot will not be broadcast by the base station until the next slot, lookahead WWP exploits the idle slot in between and initiates a new contention using an estimated window, without waiting for the contention information of the current slot to be available.

Intuitively, the lookahead technique reduces the number of contention slots by making use of the time waiting for broadcast from the base station. Each mobile station does not wait for the result of contention of the previous slot to be available before setting the next window. Instead, each mobile station sets the next window based on an estimated channel state and proceeds immediately. The best case happens when every estimation is correct; in this case, only half of the slots needed by the original WWP are sufficient to resolve collision. The worst case happens when every estimation is wrong: the same number of contention slots are needed as in the original WWP. Performance improvements due to lookahead are shown in Section 4.

2.4. Analysis of Inter-Channel Access Delays

Our experimental results in Section 4 show that our proposed window protocol performs very well, with an inter-channel access delay that is geometrically distributed (or exponentially distributed when the number of stations is large). In this section, we present theoretical justifications of this behavior.

Theorem 1. Assume the following conditions. (a) There are N contending stations. (b) S, the number of slots to resolve contentions in a contention period, is geometrically distributed with density $P(S = s) = (1-p)^{s-1}p$, $s = 1, 2, \ldots$ (c) Stations generate their contention parameters randomly so that each station has probability $\frac{1}{N}$ of being the station with the smallest contention parameter in a contention period (thereby winning the contention using the window protocol). Then X, the number of contention slots elapsed between two consecutive successful accesses of the channel by the same station, is geometrically distributed with density

$$P(X = x) = \left(1 - \frac{p}{N}\right)^{x-1} \frac{p}{N} \qquad x = 1, 2, 3, \ldots \tag{4}$$

Proof. Let X be made up of R contention periods, where the i'th contention period, $1 \le i \le R$, requires S_i contention slots. Therefore, $X = \sum_{i=1}^{R} S_i$. Since $S_1, \ldots, S_R$, R are independent, and $S_1, \ldots, S_R$ are nonnegative integers with a common density, the probability generating function of X is given by

$$\Phi_X(t) = \Phi_R(\Phi_{S_t}(t)) \qquad -1 \le t \le 1 \tag{5}$$

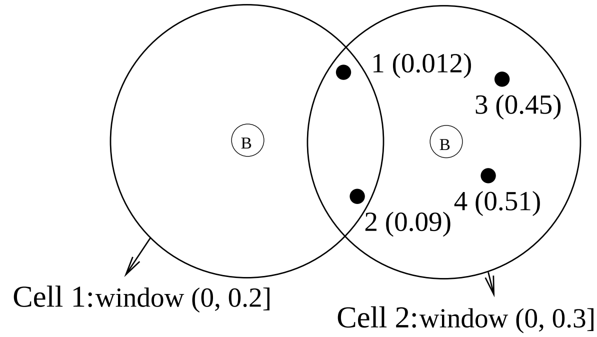


Figure 2: Impact of mobile stations in an overlapped area.

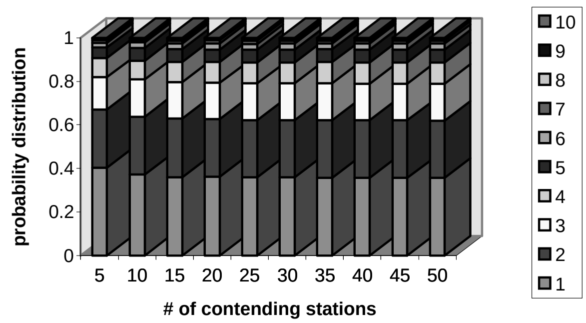


Figure 3: Distribution of number of contention slots versus number of contending stations in each contention period in one cell under heavy load.

where $\Phi_X(t) = \sum_{x=0}^{\infty} P(X = x)t^x$. It is easy to show that $\Phi_S(t) = \frac{tp}{1-t(1-p)}$ and that $\Phi_R(t) = \frac{t/N}{1-t(1-1/N)}$. Substituting these terms into (5) and simplifying the equation, we have

$$\Phi_X(t) = \frac{tp/N}{1 - t(1 - p/N)} \tag{6}$$

which is exactly the probability generating function of (4). ■

3. WIRELESS WINDOW PROTOCOL FOR MULTIPLE CELLS

As is discussed in Section 1, a station in an overlapped area between two (or more) cells may not be able to receive broadcast information reliably from its assigned base station (since base stations use the same frequency in their downlinks). As a result, it will not be able to update its window bounds when contention information broadcast by the base station is lost. Similarly, a base station may receive incorrect contention information when a mobile station in an overlapped area, but belonging to another cell, contends to use the uplink. To cope with these problems, the basic WWP needs to be modified.

3.1. Methods to Resolve Collisions in Overlap Areas

To illustrate the problems, consider the scenario in Figure 2. Cells 1 and 2 are adjacent to each other. Stations 1 and 2 are in Cell 1 initially and migrate into the overlapped area, and

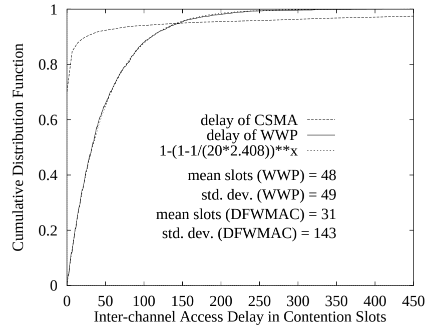
Stations 3 and 4 are in the coverage of Cell 2. Suppose four stations begin a contention period simultaneously using contention parameters shown in parentheses in the figure. Assume that the window of stations in Cell 1 is $(0, 0.2]$, and that in Cell 2 is $(0, 0.3]$. (The windows can be different due to different load estimations in the two cells.) Since Stations 1 and 2 have parameters inside the window of Cell 1, they transmit. Stations 3 and 4 refrain from transmission as their parameters are out of the window of Cell 2. However, the base station in Cell 2 hears collision in the uplink because of the transmission by Stations 1 and 2. Since it cannot tell whether the collision is caused by stations in its own cell or by stations on the boundary of an adjacent cell, it broadcasts the collision state, causing Stations 3 and 4 to reduce their windows further in subsequent intervals and never getting a chance to transmit.

We have studied two mechanisms to address this problem. The first mechanism uses a *relaxed upper bound* so that $\mathcal{U}$ is not reduced to $\mathcal{W}$ after collision is detected in window $(\mathcal{L}, \mathcal{W}]$. The basic idea is that, if the collision information broadcast by a base station is incorrect due to interference from stations in adjacent cells, then reducing the upper bound to $\mathcal{W}$ is incorrect, and the bounds will need to be set to $(\mathcal{W}, \mathcal{U}]$ in the future. Simulations show that relaxing the upper bound may still result in stations being excluded from transmission when incorrect collision information is broadcast repeatedly by the base station.

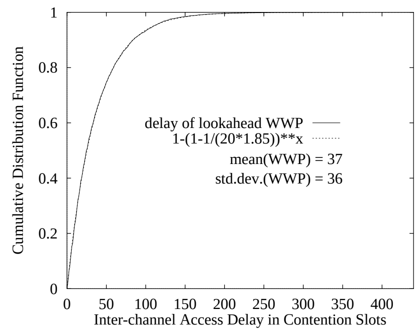
The second mechanism we have studied is *bounded contention*. It requires a base station to keep track of the number of contention slots elapsed in the current contention period. If this number exceeds a threshold, then the base station assumes that interference has caused incorrect window updates, and terminates the current contention period by broadcasting a success message in its downlink. Since the chance for the same interference in successive contention periods is very small, the scheme will eventually resolve contentions in the use of the channel. We show the performance of this scheme in Section 4.

3.2. Discussions

There are two differences between WWP and the corresponding window protocol developed for wire-based Ethernets [2]. First, the information broadcast by a base station in the downlink in one cell may be corrupted by broadcasts by base stations in adjacent cells, preventing stations in the overlap area of two cells to update their windows correctly. We have discussed modifications to WWP to cope with this situation in Section 3.1. Second, stations on Ethernets can listen while transmitting and can stop transmission immediately after detecting collisions. Hence, contention and collision detection can be carried out concurrently. In contrast, in wireless LANs, mobile stations rely on the base station to broadcast the state of contention in the second half of a contention slot. As a result, the uplink and downlink are idle half of the time. We utilized the idle time by looking ahead and testing the contention state using a different window, without waiting for the contention state of the current window to be available.



(a) DFWMAC versus WWP



(b) Lookahead WWP

Figure 4: Distributions of inter-channel access delays for a population of 20 stations. Only the distributions of one of the stations are plotted. The distributions of the remaining stations are similar.

4. PERFORMANCE EVALUATION

We have carried out simulations to evaluate WWP's performance and compare it to the DCF part of DFWMAC, the draft IEEE 802.11 standard. We have written our simulator in CSIM [5], a discrete event process-oriented simulation library. We evaluate the performance under heavy load, namely, every station always has a message ready to send. Performance is evaluated by the number of contention slots to resolve the use of the channel and the inter-channel access delay by the same station.[1] The simulations were run until the 0.95 confidence interval is reached for each station.

Figure 3 shows the distribution of the number of contention slots required by WWP in each contention period in the one-cell case. The distribution is independent of the number of contending stations, resulting in an average of around 2.4 contention slots. Although this load-independent behavior is common in other schemes [3], WWP is a random-access scheme that allows new stations to join at any time (as opposed to ordered-access schemes that require new arriving stations to wait for all existing stations to transmit before joining). It also has much better delay distribution in successive accesses to the channel by the same station as compared to other random-access schemes. This is discussed next.

Figure 4a shows the delay distribution of successive access of the channel by the same station in a population of 20 stations. DFWMAC has very skewed inter-channel ac-

[1] The duration of a contention slot depends on the transmission speed, the length of a packet, and the mechanism to detect collision.

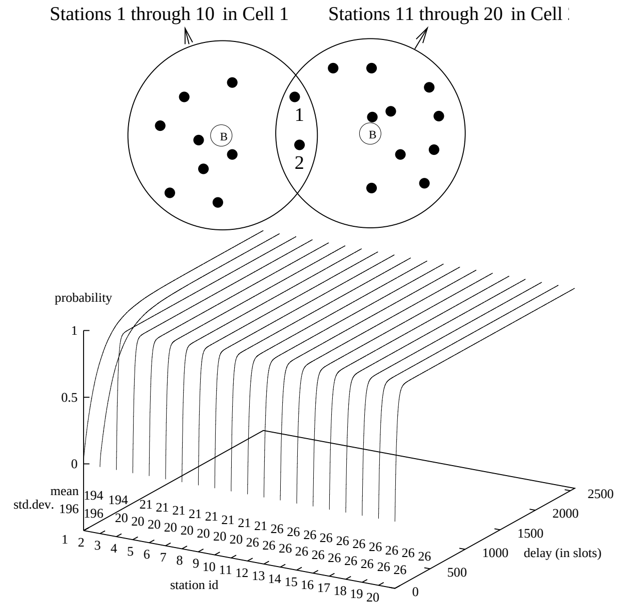


Figure 5: (a) Configuration of mobile stations in two cells (b) Performance of bounded contention WWP in two cells

cess delays: over 85% of the accesses can be made within a few contention slots, while 5% of the accesses require exceedingly long delays. This is obviously undesirable as far as fair allocation is concerned. Moreover, such behavior may make it difficult for higher-level protocols to maintain certain quality-of-service (QoS) requirements for applications. In contrast, the inter-channel access delay of WWP is geometrically distributed with an average of 48.2 contention slots. This corresponds closely to the analytical distribution shown in Theorem 1 with $p = \frac{1}{2.408}$ and $N = 20$. The advantage of the geometric distribution is that it is memoryless: every station has the same chance to access the channel, independent of the station that just accessed the channel successfully. This is better than DFWMAC that gives preference to stations that just accessed the channel successfully.

Figure 4b shows the inter-channel access-delay distribution under heavy load using the lookahead method described in Section 2. It shows that lookahead can reduce contention delays by 23%.

Finally, we evaluate the performance of the bounded contention algorithm introduced in Section 3 to handle stations in the overlapped areas of multiple cells. We carried out our experiments using the two-cell configuration in Figure 5a. We assume that each cell has 10 mobiles, numbered consecutively $1, 2, \ldots, 10$ in Cell 1 and $11, 12, \ldots, 20$ in Cell 2. Mobiles 1 and 2 are in the overlapped area of the two cells.

Figure 5b plots the inter-channel access-delay distribution for each station in the two cells using WWP with bounded contention. Stations 3 through 10 in Cell 1 have similar delay distributions, whereas Stations 11 through 20 in Cell 2 have similar delay distributions. The average delays and delay deviations are slightly larger for stations in Cell 2 due to interference from Stations 1 and 2 in Cell 1. The average delays of Stations 1 and 2 are larger than those of the other stations in Cell 1 because these stations are in the range of both bases.

The skewed access pattern of DFWMAC still exists in the multi-cell scenario and is not shown again in this section.

Our results show that, even in the presence of stations in the overlapped areas of multiple cells, the inter-channel access delay of WWP is geometrically distributed and is very close to that of stations in a single-cell scenario.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have described the design and performance evaluation of WWP in the one-cell and multiple-cell scenarios. WWP is designed to address the problems persistent in wireless random-access methods, *i.e.*, poor short-term fairness, and the hidden- and exposed-terminal problem. Our analytical and experimental results have confirmed that WWP is an efficient, scalable and fair protocol as compared to DFWMAC.

Our future work involves building a prototype of WWP in which stations communicate with wireless modems that can detect three possible states of a channel: no transmission, success, and collision. The effects of WWP on higher level protocols need to be studied as well.

6. REFERENCES

[1] B.S.Tsybakov and V.A.Mikhailov. Random multiple packet access: part-and-try algorithm. *Problems of information transmission*, 16(4):305-317, October 1980.

[2] B.W.Wah and J.Y.Yuang. Resource scheduling for local computer systems with a multiaccess network. *IEEE Trans. on Computers*, C-34(12):1144-1157, December 1985.

[3] D.Bertsekas and R.Gallager. *Data Networks.* Prentice-Hall, 1992.

[4] J.Weinmiller, H.Woesner, J.P.Ebert, and A.Wolisz. Analyzing and tuning the distributed coordination function in the ieee 802.11 DFWMAC draft standard. In *Int. Workshop on Modelling, Analysis and Simulation of Computer and Telecommunication Systems*, February 1996.

[5] Mesquite Software Inc. *CSIM17 User Guide - C Version*. available at http://www.mesquite.com/.

[6] R.G.Gallager. Conflict resolution in random access broadcast networks. In *AFOSR Workshop Communication Theory and Applications*, September 1978.

[7] Xiao Su. *Design and Evaluation of a Window-Based Wireless Medium Access Control Protocol*. M.Sc. Thesis, Dept. of Computer Science, Univ. of Illinois, Urbana, IL, May 1997.

[8] V.Bharghavan, A.Demers, S.Shenker, and L.Zhang. MACAW : A media access protocol for wireless LANs. In *ACM SIGCOMM Conf. on Communications Architectures, Protocols and Architectures*, September 1994.