A Distributed Medium Access Protocol for Wireless LANs

R. Ellen Khayata, Carlos M. Puig, and Jonathan M. Zweig
Apple Computer, Inc.
1 Infinite Loop, MS: 301-4J
Cupertino, CA 95014
ellen@apple.com

Abstract

The Urn protocol has been proposed for distributed access to the radio medium. Access rights to the channel are determined at each station through estimation of the total network load. The Urn's performance was shown to be similar to that of Slotted Aloha at light load, while defaulting to TDMA-like behavior under heavy load. We simulated the Urn protocol in a realistic radio environment, taking into account channel effects such as fading, capture, collisions, hidden nodes, and noise. We quantify the resulting disparity of traffic load estimation across users. We propose techniques for reducing this divergence. We also extend the throughput-delay results to include peer-to-peer and centralized server situations, through nonuniform traffic and priority considerations. We show the throughput of the Urn scheme to be high under all these effects. Introducing two classes of priority service reduces the average delay experienced by the high priority users.

I. Introduction

Medium access protocols for the radio channel often fall into two extremes: totally distributed versus centralized. Distributed schemes are favored in ad-hoc networks, where portable computers users could meet spontaneously to exchange documents and movies. Centralized schemes are preferred for wireless access to a wired network through a Base Station. The Base Station controls access to the channel, and may reserve bandwidth for time-bounded traffic. Both approaches are valuable. Ad-hoc LAN users will want to access the wired backbone network to retrieve files or search libraries. Base Stations will not always be available and introduce unnecessary overhead for relaying packets between nearby stations.

In [1], Kleinrock and Yemini propose the Urn approach, which allows for collision avoidance in a slotted, completely distributed environment. At the beginning of each time slot, each station allocates access rights based on limited network load information gleaned from passing traffic. For example, the traffic load can be estimated from listening to the acknowledgements, or monitoring the channel for short reservation packets. Given that estimate, the algorithm maximizes the probability that only one station sends in a time slot. In [1], Kleinrock and Yemini assume that the users agree as to the number of busy users, and so all users agree on the identity of those given access rights in a particular slot. They show the throughput to be high at low and medium loads, and to reach TDMA at high loads. The delay at low loads is low, and settles to that expected in a TDMA environment at high loads. Some researchers have extended this work by suggesting breaking up the contention times into TDMA and Urn [2] or dropping selected packets [3].

We extended those results by simulating the Urn protocol in a realistic radio environment, taking into account channel effects such as fading, capture, collisions, hidden nodes, and noise. These radio effects impact the performance of the basic Urn protocol significantly, because they introduce disagreements on the assignment of access rights in each slot. We quantify the resulting decrease in throughput and increase in delay. We then propose and simulate simple techniques for reducing those effects. Although the scheme was originally proposed as a purely decentralized scheme, we extended it to support LANs with Base Stations through the use of priority classes and nonuniform traffic distributions.

We show that the throughput-delay performance of the Urn scheme is robust under all these networks configurations. Introducing two classes of priority service reduces the average delay for the high priority users, relative both to the single-priority case and to the lower priority stations in the two-priority case. Thus priority can be used to support time-bounded services such as voice and video without having to reserve bandwidth.

In section II, we briefly present the basic Urn algorithm, as defined in [1,2]. In section III, we discuss the sources of degradation, and offer solutions. In section IV, we describe the simulation model and parameters. In section V, we first show the performance degradation resulting from applying a realistic channel model to the basic Urn scheme. We then show the throughput-delay improvement from implementing the adjustments described in section III. We also show the performance of the hybrid ad-hoc and Base Station LAN modes with dual-priority. We present our conclusions in section VI.

1058-6393/95 $4.00 © 1995 IEEE

II. Basic Urn algorithm

Let

N = number of stations in the network.

n = number of busy stations (stations with a packet to send); $n \leq N$.

To maximize the probability that only one station sends in a time slot, the number of stations that should be given permission to transmit in a time slot has been shown to be k = int $[N+1/n]$.

Each station needs to determine:

1) Number of busy stations (n). This information can be estimated either solely from the acknowledgement statistics, or by also requiring busy stations to send out a short reservation packet before each data packet.

The first approach introduces no overhead, but assumes quasi-static traffic loads and requires a station to listen long enough to gather statistically significant estimates. Estimates of future loads can be improved by numbering packets and including previews of coming traffic (e.g. a packet could be identified as number 4 out of 100 in a file transfer.)

2) The identity of the stations being given permission to transmit in this time slot. Once the number (k) of stations to be allowed to transmit in a certain slot has been computed, determining who gets to send can be done in either of two ways:

A) Each station has a pseudorandom number generator with the same seed as everyone else. Each station draws k numbers between 1 and N without replacement. The station transmits (if it has something to send) when its number is selected.

B) Imagine the station numbers equally spaced around a circle. A window of size k moves around the circle, enabling everyone within it to transmit. If there is a collision, the window size is reduced in subsequent slots until a packet gets through. This round-robin approach allows bounds the maximum delay to N slots.

III. Discussion of the Urn's mechanics

Each time slot allows for a short reservation packet, a data packet, and an acknowledgement packet. All stations which correctly hear a reservation packet increment their estimated number of busy stations (n) by one. Otherwise, if they hear a collision, they increment their n by two. A station with an access right and a ready data packet sends the packet. The destination sends the source an ack packet for a correctly received data packet. All stations which hear the ack packet decrement their n by one.

The algorithm is designed to maximize the probability that exactly one user sends in a time slot. This algorithm will occasionally assign access rights to more than one busy station, resulting in a collision. It will occasionally assign access rights to stations with no packets ready, resulting in an empty slot. Either situation introduces inefficiencies, but neither is very likely to occur. Collisions are most likely under heavy loads, but the Urn scheme eliminates collisions by using TDMA at high loads. Empty slots are most likely under light loads, when wasting a time slot is affordable.

Ack collisions cause stations to overestimate n. These collisions are infrequent because acks only occupy a short time of what must have been a favorable channel for both the data packets.

Suppose all stations start off with the same pseudo-random number generator seed. At each time slot, different users will have different n estimates because of hidden nodes, fading, and collisions. Stations may then differ on the number of access rights k for that slot. By not using their random number generators equally, stations will end the slot with different seeds. This problem can easily be solved by requiring all stations use their generators a certain fixed, large number of times, but only grant access to the first k stations they chose. Stations might still differ as to k, but will agree on the addresses of $\min\{k\}$ across all stations.

Radio channel effects will cause stations to have different, possibly incorrect, estimates of n, which will result in collisions. (Note that $k=1$ for all $n \geq (N+1)/2$.) Agreement can be restored either by resynchronizing on estimates of n embedded in ongoing traffic or through periodic broadcasts. Stations might agree to choose the largest n they hear. Overestimating n will lead to fewer collisions, but results in unused slots.

Implementation considerations

As with all slotted schemes, some inefficiency may result when long packets aren't integer multiples of the slot size or when not enough short packets can be grouped into the fixed slot size. Slotting also implies clock synchronization among the users, which can be realized using well-known techniques.

New users may choose their IDs by listening to the traffic, and selecting an unused (or "greater") ID. When a user moves out of a network without notifying the other users, permission will still be given to that user, which may result in some inefficiency.

In a radio environment, a station sending out a reservation cannot hear its own reservation packet collide with another. These stations then underestimate n, since they increment their n by one. This can easily be remedied by forcing reserving stations to increment their number of busies by two if they don't hear an ack back from their destinations. This heuristic improved the performance significantly.

Base stations and time bounded services are supported through priority classes. Higher priority classes are given proportionally more access rights. The urn scheme allows for a fixed average delay at high loads, making it easy to support time-bounded services.

IV. Simulation model and parameters

Parameters common to all simulations

Total channel capacity: 20 Mbps

Network layout:

.14...15...16...17...18...19...20..
....
..7....8....9...10...11...12...13..
....
..0....1....2....3....4....5....6..

Total number of users in the network, N: 21
Regular network case:
Horizontal distance between adjacent cells: 5 m
Vertical distance between adjacent cells: 3 m
Hidden node case:
Horizontal distance between adjacent cells: 15 m
Vertical distance between adjacent cells: 3 m

Hidden nodes were introduced by separating the nodes horizontally. In our simulations, nodes 0,7, and 14 were hidden from nodes 6,13, and 20. The Base Station/server is hidden from no one.

Servers and heavy users: In the above diagram, node 10 is the server when there is only one server in the network. Two Base Stations/servers, or two heavy users, are located at nodes 8 and 12.
Ratio of server's traffic to each nonserver node's traffic: 5
Priority of server relative to regular users: 5
Ratio of heavy user traffic to each regular user's traffic: 5
Priority of heavy users relative to regular users: 1 (same priority)

Packet sizes:
Total Slot time: 295 ms
Reservation packet length: 70 bytes
Ack packet length: 50 bytes
Fixed overhead on each data packet: 50 bytes
Total data packet size: 550 bytes, including overhead.

Radio characteristics:
Transmit power: 30 dBm
Minimum received signal strength: -70 dBm
Receiver noise level: -85 dBm
Minimum SNR or SIR for correct reception: 15 dB
No transmitting or receiving antenna gain is assumed
Two-level switched diversity
Center Frequency: 5 GHz
Power loss exponent is 2 until 8.5 m, and 3.6 beyond that.
Rayleigh fading.

Maximum queue length at each user: 40 packets
Maximum number of data packet retries if no ack is received: 16

Information level in the network:
Info 1 corresponds to perfect information across all stations. All stations use the actual (exact)) value of *n*, regardless of collisions, etc.
Info 3 corresponds to the case where the number of busy stations is incremented by 0, 1, or 2. In this case, the reserving station only increments the number of busy stations by one, unless an acknowledgement is not received for the first transmission attempt, when a collision is assumed, in which case the count is incremented by two.

V. Simulation results

A. No servers

Fig. 1a Throughput No servers, Info 1, 2 Antennas

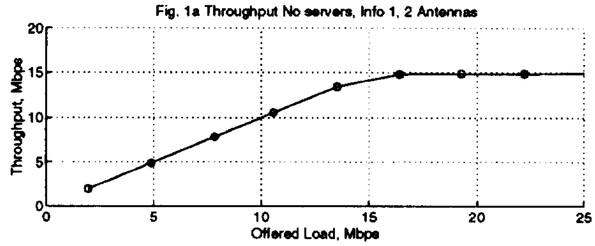
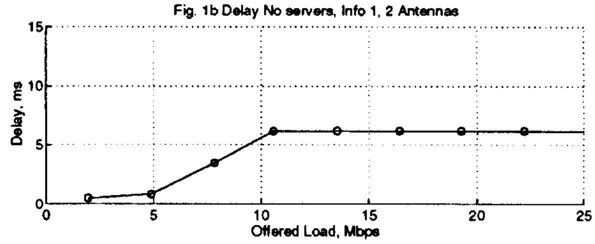


Fig. 1b Delay No servers, Info 1, 2 Antennas



No servers, Information level 1, Two Antennas
Figures 1a and 1b show the throughput and the transfer delay versus offered load, respectively under conditions of perfect information at all nodes. All users agree on the identity of busy users in each slot. As expected, the throughput increases linearly with increasing offered load. It saturates at 15 Mbps. The transfer delay increases with offered load, and reaches the TDMA level of 6 ms, which is equal to 21 slot times. These results agree with those presented by Kleinrock and Yemini [1].

Fig. 2a Throughput No servers, Info 3, 2 Antennas, No seed fix

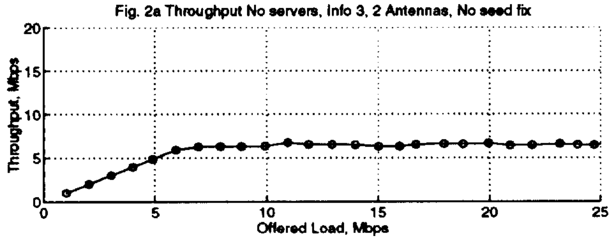
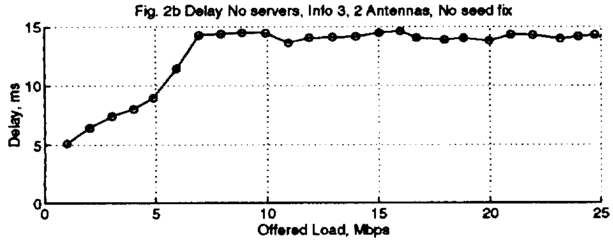


Fig. 2b Delay No servers, Info 3, 2 Antennas, No seed fix



No servers, Information level 3, Two Antennas, No seed fix

Figures 2a and 2b show the degradation in performance when no seed fix is included with independent estimates of n. The throughput settles at unacceptable levels.

Fig. 3a Throughput No servers, Info 3, 2 Antennas

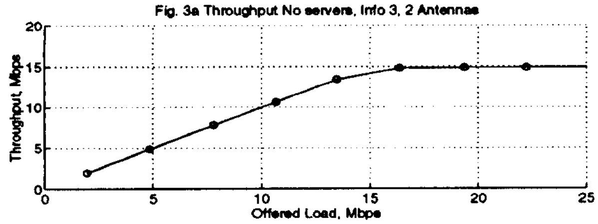
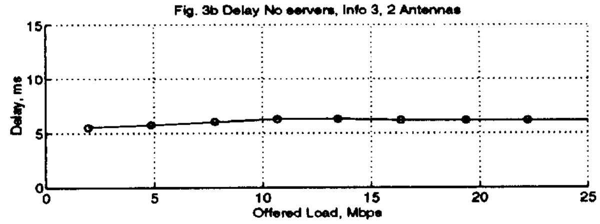


Fig. 3b Delay No servers, Info 3, 2 Antennas



No servers, Information level 3, Two Antennas

Figures 3a and 3b show the effects of imperfect information, when stations exchange no information. The throughput is nearly the same as in the perfect information case. Performance improves because: (1) all stations begin each time slot with the same pseudo-random number generator seed by invoking the generator a large, fixed number of times in each slot; and (2) the reserving station increments its n by two if it does not hear an ack in response to its data packet. The second procedure leads to a slight overestimate in n.

Fig. 4a Throughput No servers, Info 3, 2 Antennas, Hidden nodes

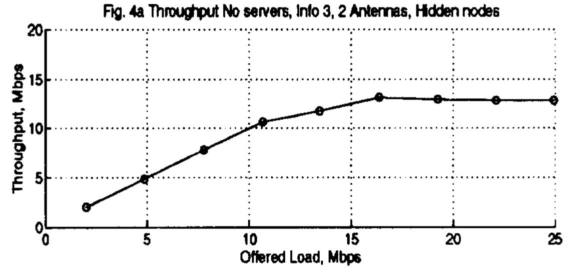
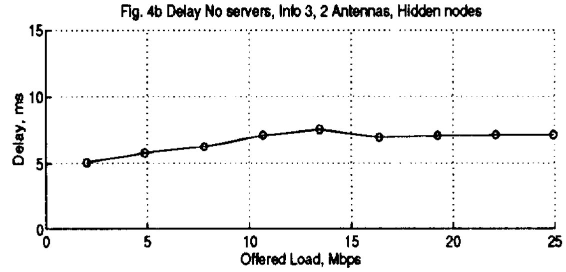


Fig. 4b Delay No servers, Info 3, 2 Antennas, Hidden nodes



No servers, Information level 3, Hidden nodes, Two antennas

Figures 4a and 4b show the effect of hidden nodes. Two columns of three stations, located at the right and left extremes of the network, are hidden from each other. At high loads, the throughput reaches 13 Mbps, and the transfer delay settles at 7 ms. The bump in the delay near 12 Mbps is due to the excess collisions present in a system with many hidden nodes.

B. Base Stations/Servers

Although the Urn scheme is a completely distributed scheme, it can easily be extended to include priority for servers or time-bounded services. In our simulations, we assumed that the servers are more frequently sources and destinations than other nodes, and have proportionally greater access rights. This priority scheme leads to higher throughput and lower delay for the higher priority server traffic. Note that higher priority traffic does not always win access rights over the lower priority traffic. Thus, high priority classes do not provide a guaranteed quality of service (QOS), but only a best effort model. QOS cannot be guaranteed in highly variable environments such as the radio channel.

In the simulations, servers do not send packets to each other, since they are likely to be connected via a wired network. We chose the servers to be sources and destinations five times more often than regular users.

Fig. 5a Throughput One server, Info 3, 2 Antennas

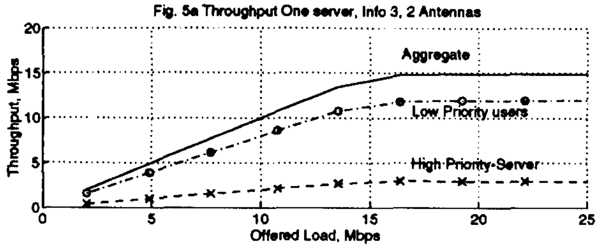
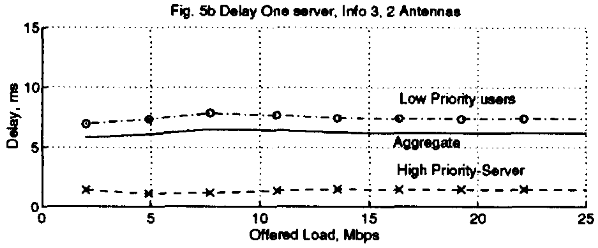


Fig. 5b Delay One server, Info 3, 2 Antennas



One server, Information level 3, Two Antennas

Figures 5a and 5b show the effect of imperfect information on the performance: the throughput is robust, staying high at all times, while the delay suffers, as the average delay settles to 6 msec at even low offered load. Note that the server throughput is lower because the servers constitute only a fraction of the total offered load.

Fig. 6a Throughput Two servers, Info 3, 2 Antennas

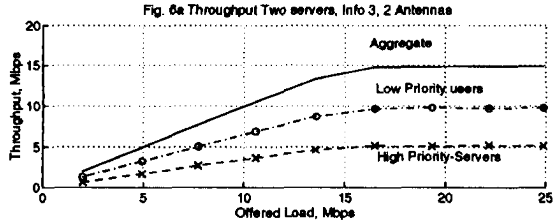
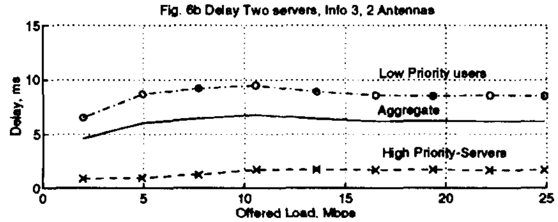


Fig. 6b Delay Two servers, Info 3, 2 Antennas



Two servers, Information level 3, Two Antennas

Figures 6a and 6b show the effect of having two high priority servers in the network. Our model usually includes one server per network, but two independent servers might occur. The bump in the delay is due to the servers' estimates of n being smaller than the low-priority stations. The two servers' traffic comprises a greater fraction of the total offered load, so that they account for a third of the throughput. The transfer delay is still under 2 msec for the servers, and close to 8 msec for the non-servers.

C. Two heavy users

Fig. 7a Throughput Two heavy loads, no servers, Info 3, 2 Antennas

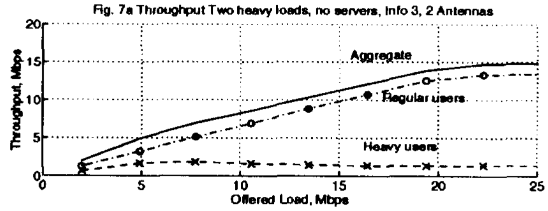
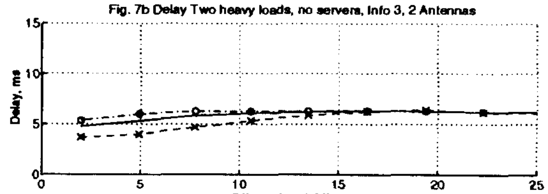


Fig. 7b Delay Two heavy loads, no servers, Info 3, 2 Antennas



Two heavy users, Information level 3, Two Antennas

Figures 7a and 7b show the performance when two users offer five times as much traffic as the average user, but do not receive higher priority. Like the servers, the two heavy users are located at nodes 8 and 12. Comparing Figs 7a and 7b with 6a and 6b, we see that the heavy users' throughput is considerably lower in the no priority case, and that their delay quickly settles to that of the average users. The delay of heavy stations' traffic is better than the that of other stations' at low loads; possibly because they can dominate the channel when there is little competition. The severely nonuniform traffic distribution degrades the throughput slightly at higher loads.

VI. Conclusions

Our simulation encompasses many realistic features, including radio propagation effects (fading, capture, and hidden nodes), difference in estimates across nodes, different priority classes (Base Stations/servers, time-bounded services). The modified Urn scheme provides high throughput and low delay even under those conditions. In the imperfect information case, the delay settles to TDMA levels at relatively low offered load, but the throughput remains high. The delay seen by higher priority traffic is substantially lower than that seen by lower priority users.

References

[1] L. Kleinrock and Y. Yemini, "An Optimal Adaptive Scheme for Multiple Access Broadcast Communication," International Communications Conference (ICC) 1978, p.7.2.1-7.2.5.

[2] E.D. Sykas, D.E. Karvelas, and E.N. Protonotarios, "Combined Urn and TDMA Scheme for Multiple-Access Protocols," Computer Communications, 1983, V6, N4.

[3] K K. Mittal, and A. N. Venetsanopoulos, "On the Dynamic Control of the Urn Scheme for Multiple Access Broadcast Communication Systems," IEEE Trans on Communications, Vol COM-29, No. 7, July 1981, pp. 962-970.