On the Scalability of IEEE 802.11 Ad Hoc Networks

Lifei Huang
huang.343@osu.edu

Ten-Hwang Lai
lai@cis.ohio-state.edu

The Ohio State University
Department of Computer and Information Science
2015 Neil Avenue
Columbus, OH 43210-1277

ABSTRACT

The IEEE 802.11 standards support the peer-to-peer mode Independent Basic Service Set (IBSS), which is an ad hoc network with all its stations within each other's transmission range. In an IBSS, it is important that all stations are synchronized to a common clock. Synchronization is necessary for frequency hopping spread spectrum (FHSS) to ensure that all stations "hop" at the same time; it is also necessary for both FHSS and direct sequence spread spectrum (DSSS) to perform power management. This paper evaluates the synchronization mechanism, which is a distributed algorithm, specified in the IEEE 802.11 standards. By both analysis and simulation, it is shown that when the number of stations in an IBSS is not very small, there is a non-negligible probability that stations may get out of synchronization. The more stations, the higher probability of asynchronism. Thus, the current IEEE 802.11's synchronization mechanism does not scale; it cannot support a large-scale ad hoc network. To alleviate the asynchronism problem, this paper proposes a simple modification to the current synchronization algorithm. The modified algorithm is shown to work well for large ad hoc networks.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms

Algorithms, performance

Keywords

IEEE 802.11, ad hoc networks, scalability, clock synchronization

MOBIHOC'02, June 9-11, 2002, EPFL Lausanne, Switzerland.

1. INTRODUCTION

The IEEE 802.11 standards [4] are widely used for wireless LAN (WLAN). They support two types of WLAN: one in infrastructure mode and the other in ad hoc mode. In the infrastructure mode, a station serves as the access point and communications can only take place between the access point and the other members of the basic service set (BSS). When the infrastructure does not exist or does not work, the ad hoc mode is useful. In the IEEE 802.11 standards [4], an ad-hoc network is called an *Independent Basic Service Set* (IBSS), in which all of the stations are within each other's transmission range.

Performance analysis of IEEE 802.11 WLAN has been reported in [2, 1, 7, 10]. Reference [2] evaluates the performance of the Distributed Coordination Function (DCF) of IEEE 802.11 standards and proposes an adaptive contention window protocol to replace the exponential backoff protocol. In [1] and [10], the saturation throughput of DCF is analyzed using different techniques. The effect of network size and traffic patterns on the capacity of ad hoc wireless networks is examined in [7], where the locality of traffic is shown to be a key factor of scalability of ad hoc networks.

All the above works are focused on the IEEE 802.11 MAC protocol. However, the scalability issue has not been fully studied. Envision that in a classroom or in an auditorium, more than 100 students turn on their laptops to form an IBSS. Will the ad hoc network work properly? This question is not trivial. In this paper we will study the scalability of IEEE 802.11 ad hoc networks from the viewpoint of clock synchronization. To the best of our knowledge, this paper is the first to report such a study.

In an IEEE 802.11 WLAN, each station maintains a clock or timer. (A clock is called a timer in [4]; therefore, we use timer and clock interchangeably in this paper.) Clock synchronization is needed for power management, synchronization of frequency hopping, and coordination of Point Coordination Function (PCF). In power management each station uses its clock to determine the beginning and the end of the *ad hoc traffic indication* (ATIM) window. In FHSS, each station determines when to "hop" to a new channel according to its timer. Finally, in an IEEE 802.11 infrastructure network that uses PCF, the timer is used to predict the start of a super frame.

In order to synchronize the clocks, IEEE 802.11 specifies the *timing synchronization function* (TSF). TSF is easy for infrastructure networks. The access point periodically broadcasts a beacon frame that contains the timing information,

	FHSS	DSSS	Infrared
aCWmin	15	31	63
aSlotTime	$50\mu s$	$20\mu s$	$8\mu s$

Table 1: Beacon generation window and slot time

and all the other stations that receive the beacon frame adopt the access point's timer value. In an IBSS, since there is no access point, a distributed algorithm is employed for timing synchronization. Basically, timing information is exchanged through periodically transmitted beacon frames by every station. A station adopts the timing in a received beacon if the received time is later than the station's own TSF timer.

This paper evaluates the beacon generation procedure specified in the IEEE 802.11 standards. We show that due to beacon contention, the stations may fail to successfully transmit beacon frames. As a result, some stations in the IBSS may become so out of synchronization with others that power management or FHSS can not work properly. By both analysis and simulation, we show that when the number of stations in an IBSS is not very small, there is a non-negligible probability that stations may get out of synchronization. The more stations, the higher probability and the larger ratio of asynchronism. Thus, the current IEEE 802.11's synchronization mechanism does not scale well; it cannot support a large-scale ad hoc network, say, from 150 to 200 stations as in the classroom example mentioned earlier. As to be seen, even when the network size is a moderate 30, the fastest clock in the IBSS already suffers from asynchronism with a high probability.

To alleviate the asynchronism problem, we propose a simple modification to the current IEEE 802.11 synchronization algorithm. With the modification, the synchronization algorithm is observed to work well for large ad hoc networks.

The rest of the paper is organized as follows. The next section reviews the clock synchronization protocol as specified in the IEEE 802.11 standards. Section 3 analyzes the beacon generation scheme, while Section 4 analyzes the phenomenon of clock asynchronism. Numerical and simulation results are presented in Section 5. An adaptive timing synchronization procedure is proposed in Section 6 as a remedy for the asynchronism problem. Section 7 concludes the paper.

2. CLOCK SYNCHRONIZATION

This section reviews the Timing Synchronization Function (TSF) as specified in the IEEE 802.11 specifications [4], and comments on a few related clock synchronization algorithms.

2.1 The TSF of IEEE 802.11

MAC management plays an important role in IEEE 802.11 ad hoc networks. Synchronization functions and power management functions are implemented in this sublayer.

According to the IEEE 802.11 specifications [4], each station maintains a TSF timer (clock) with modulus 2^{64} counting in increments of microseconds (μs). Clock or timing synchronization is achieved by stations periodically exchanging timing information through beacon frames, which contains a timestamp among other parameters. Each station in an IBSS shall adopt the timing received from any beacon that has a TSF time value (the timestamp) later than its own TSF timer. All stations in the IBSS adopt a common value, aBeaconPeriod, that defines the length of beacon intervals or periods. This value, established by the station that initiates the IBSS, defines a series of *Target Beacon Transmission Times* (TBTTs) exactly aBeaconPeriod time units apart. Time zero is defined to be a TBTT. Beacon generation in an IBSS is distributed; all stations in the IBSS participate in the process as follows.

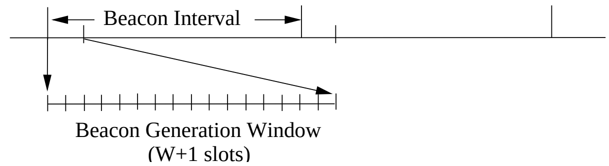

Figure 1: Beacon generation window.

Beacon Generation and Clock Synchronization:

1. At each TBTT each station calculates a random delay uniformly distributed in the range between zero and 2·aCWmin·aSlotTime. (The aCWmin and aSlotTime parameters are specified in Table 1.)
2. The station waits for the period of the random delay.
3. If a beacon arrives before the random delay timer has expired, the station cancels the pending beacon transmission and the remaining random delay.
4. When the random delay timer expires, the station transmits a beacon with a timestamp equal to the value of the station's TSF timer[1].
5. Upon receiving a beacon, a station sets its TSF timer to the timestamp of the beacon if the value of the timestamp is later than the station's TSF timer[2]. (It is important to note that *clocks only move forward and never backward*.)

Thus, as illustrated in Fig 1, at the beginning of each beacon interval, there is a *beacon generation window* consisting of $W+1$ slots each of length aSlotTime, where $W = 2\cdot$aCWmin. Each station is scheduled to transmit a beacon at the beginning of one of the slots. For FHSS, the beacon size is at least 550 bits. Therefore, for the data rate of 1Mbps the beacon length is 11 slots. The beacon length is 7 slots if the data rate is 2Mbps.

The basic idea of power management is to turn off the station whenever it doesn't have to be active. To prevent from missing data destined for it, each station wakes up from time to time. Thus each station switches between *power save* (PS) state and *awake* state. In an IBSS, the ad hoc traffic indication (ATIM) window determines the time period during which all the stations including those in PS mode must be awake. In the ATIM window, only beacon or ATIM frames shall be transmitted. The length of ATIM window is specified by aATIMWindow parameter. It follows a TBTT and ends when TSF timer $\bmod$ aBeaconPeriod = aATIMWindow.

[1] More precisely, the timestamp is equal to the value of the TSF timer at the time the first bit of the timestamp is transmitted to the physical layer plus the transmitting station's delay through its physical layer from the MAC-PHY interface to its interface with the wireless medium. This detail is not important for our analysis.

[2] Actually, the receiving station will adjust the timestamp by adding an amount equal to the delay through the physical layer, and the adjusted value of the timestamp is used for synchronization. This detail is not important for our analysis.

2.2 Related Work

IEEE 802.11's timing synchronization function is similar to the method proposed by Lamport [6]. With this method, it is shown in [6] that given the clock accuracy, link delay and network diameter, and assuming that a timestamp is sent *successfully* along each link at a constant frequency, the timing values of the entire network are guaranteed to be accurate to an established bound. The assumption of successful transmission of timestamp at a constant frequency is essential for the validity of the bound. This assumption does *not* hold for the TSF of the IEEE 802.11 IBSS.

A different synchronization method called reference broadcast synchronization (RBS) is presented in [3] for broadcast networks, especially for wireless ad hoc networks. A reference broadcast or beacon does not contain an explicit timestamp; instead, receivers use its arrival time as a point of reference for comparing their clocks. RBS uses nontrivial statistics methods such as regression to estimate the phase offset and clock skew of any two nodes.

Mills [8] proposes a Network Time Protocol (NTP) to synchronize clocks and coordinate time distribution in the Internet system. NTP cannot be used for sparse ad hoc networks which can be partitioned. To deal with partitioning in sparse ad hoc networks, a time synchronism algorithm is proposed in [9].

At the other end of the spectrum of ad hoc networks in terms of station population, when there are a large number of stations in an ad hoc network, as to be seen in this paper, the stations are likely to be out of synchronization.

3. ANALYSIS OF BEACON CONTENTION

As mentioned in the preceding section, at the beginning of each beacon interval, all stations in an IBSS contend with each other to send beacon frames. There are two possible outcomes of each beacon contention: at least one beacon frame is transmitted successfully or no beacon frame is transmitted successfully. For our analysis, successful transmission is defined as follows.

Definition 1. A beacon transmission is considered *successful* if it encounters no collision.

Thus, for ease of analysis, our model assumes no transmission error. This assumption is not undesirable as our analysis is intended to point out the asynchronism problem with the IEEE 802.11's TSF scheme. If transmission error is taken into account, the asynchronism problem will just be even more severe.

For an arbitrary station A, there are three possible outcomes to A's beacon contention in each beacon period:

1. Station A does not transmit any beacon because it receives a beacon before its random delay expires.
2. Station A transmits a beacon which collides with other beacons.
3. Station A transmits a beacon successfully.

We first analyze the probability that at least one station successfully transmits a beacon in a given beacon interval, and then we calculate the probability that a particular station succeeds in beacon transmission during a given beacon interval.

Suppose there are n stations. Denote the beacon generation window by $[0, W]$, which consists of $W + 1$ slots numbered 0 through W. Let $p(n, W)$ be the probability that at least one of the n stations succeeds in beacon transmission during a beacon interval. The event, E, that there is a successful beacon transmission in window $[0, W]$ consists of three disjoint sub-events:

- E_1 : There is no beacon transmission in slot 0, but there is a successful transmission in window $[1, W]$.
- E_2 : There is a successful beacon transmission in slot 0.
- E_3 : There are unsuccessful (collided) beacon transmissions in slot 0, but at least one successful beacon transmission in slot 1 through slot W.

Thus,

$$\begin{aligned} p(n, W) &= \text{Prob}(E) \\ &= \text{Prob}(E_1) + \text{Prob}(E_2) + \text{Prob}(E_3). \end{aligned} \quad (1)$$

$\text{Prob}(E_1)$ and $\text{Prob}(E_2)$ can be easily calculated as follows.

$$\text{Prob}(E_1) = \left(\frac{W}{W+1}\right)^n p(n, W-1)$$

$$\text{Prob}(E_2) = n\left(\frac{1}{W+1}\right)\left(\frac{W}{W+1}\right)^{n-1}.$$

Let $q(n, W) = \text{Prob}(E_3)$. Substituting this and the above two equations into Eq. 1 yields the following recurrence for $p(n, W)$:

$$\begin{aligned} p(n, W) = &\left(\frac{W}{W+1}\right)^n p(n, W-1) \\ &+ n\left(\frac{1}{W+1}\right)\left(\frac{W}{W+1}\right)^{n-1} \\ &+ q(n, W) \end{aligned} \quad (2)$$

The boundary condition for $p(n, W)$ is $p(0, W) = p(n, 0) = 0$.

To compute $q(n, W)$, recall that each beacon typically has a length of multiple, say b, slots. If there is a beacon collision in slot 0, the collided beacons will occupy all the slots from 0 to $b - 1$ and, as a result, there will be no successful beacon transmission before slot b. Thus, for $W > b$ and $n \geq 2$, we have the following formula for $q(n, W)$.

$$q(n, W) = \sum_{i=2}^{n}\sum_{j=0}^{n-i}\left\{C_i^n C_j^{n-i}\left(\frac{1}{W+1}\right)^i\left(\frac{b-1}{W+1}\right)^j \left(\frac{W-b+1}{W+1}\right)^{n-i-j} p(n-i-j, W-b)\right\} \quad (3)$$

where the notation C_k^n is used to denote the number of combinations of n objects taken k at a time; that is, $C_k^n = n(n-1)\cdots(n-k+1)/k!$. The term within the braces, $\{\ldots\}$, is the probability that exactly i stations transmit a beacon in slot 0, exactly j stations are scheduled to transmit beacons in slots 1 through $b-1$ (these stations will not actually transmit beacons because the medium has been occupied by the collided beacons), and at least one of the remaining $n - i - j$ stations successfully transmits a beacon in slots b through W. As for the boundary condition, if $W \leq b$ or $n < 2$, then $q(n, W) = 0$.

The value of b (i.e., the length of beacon frames in terms of slots) is calculated to be 11 for the FHSS system. For $b = 11$, the values of $p(n, W)$, calculated from Eqs. 2 and 3, are plotted as curves in Fig. 3 for two different values of W. The data

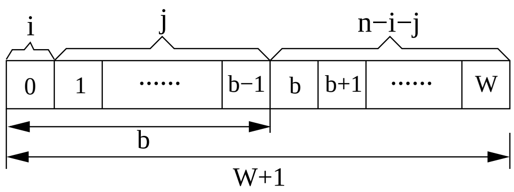

Figure 2: Explanation for p(n,W)

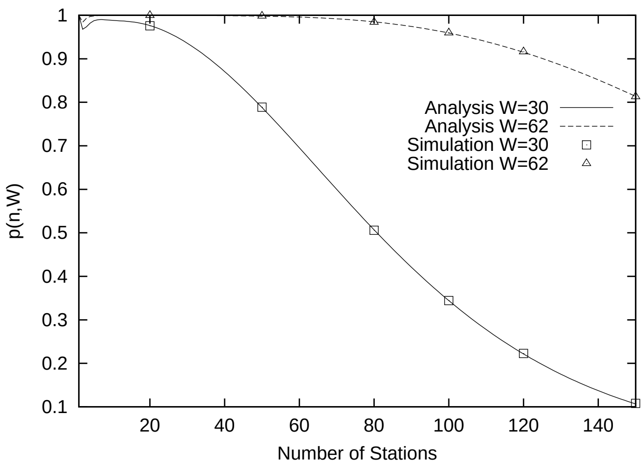

Figure 3: p(n,W) with b=11

points on the curves are generated by simulation. As can be seen, the analysis results coincide with the simulation results very well. The figure shows that given $W = 30$, $p(n, W)$ is lower than 50% when $n > 80$. That means in half of the beacon intervals, no beacon frames are transmitted successfully. As to be seen in the next section, this may cause some stations to become out of synchronization.

Now we analyze $p'(n, W)$, the probability that a particular station, say A, successfully sends a beacon in a given beacon interval. Let $p'(n, W, k)$ denote the conditional probability that station A successfully transmits a beacon given that it is scheduled to transmit in slot k. Then we have the following equation:

$$p'(n, W) = \frac{1}{W+1} \sum_{k=0}^{W} p'(n, W, k) \tag{4}$$

where $\frac{1}{W+1}$ is the probability that A is scheduled to transmit at slot k.

To compute $p'(n, W, k)$, note that $p'(n, W, k) = \text{Prob(a)} + \text{Prob(b} \wedge \text{c} \wedge \text{d)} \cdot \text{Prob(e} \mid \text{(b} \wedge \text{c} \wedge \text{d))}$, where a, b, c, d, e refer to the following events, respectively. (Figure 4 helps explain these events.)

(a) All other stations are scheduled to transmit *after* slot k. The probability of this event is $\left(\frac{W-k}{W+1}\right)^{n-1}$.

(b) No station transmits before slot i, $0 \leq i \leq k - b$.

(c) Exactly x $(2 \leq x \leq n - 1)$ stations transmit in slot i.

(d) Exactly y $(0 \leq y \leq n - 1 - x)$ stations are scheduled to transmit in slots $i + 1$ through $i + b - 1$.

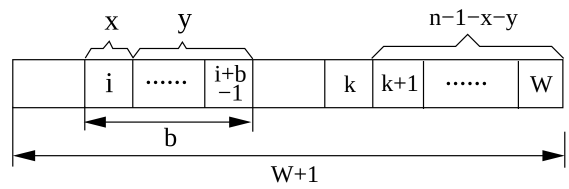

Figure 4: Explanation for p'(n,W,k)

(e) Of the $n - x - y$ stations that are scheduled to transmit within window $[i + b, W]$, the station in question successfully transmits in slot k. The probability of this event given events b, c, and d is, by definition, $p'(n - x - y, W - i - b, k - i - b)$.

The probability of event (b ∧ c ∧ d) for a particular set of values of i, x, y is

$$\begin{aligned} \text{Prob(b} \wedge \text{c} \wedge \text{d)} &= C_x^{n-1} C_y^{n-1-x} \left(\frac{1}{W+1}\right)^x \left(\frac{b-1}{W+1}\right)^y \\ &\quad \cdot \left(\frac{W-i-b+1}{W+1}\right)^{n-1-x-y}. \end{aligned}$$

Therefore, for $n \geq 3$ and $k \geq b$, $p'(n, W, k)$ satisfies the following recurrence:

$$\begin{aligned} p'(n, W, k) &= \left(\frac{W-k}{W+1}\right)^{n-1} + \sum_{i=0}^{k-b} \sum_{x=2}^{n-1} \sum_{y=0}^{n-1-x} \\ &\quad \left\{ C_x^{n-1} C_y^{n-1-x} \left(\frac{1}{W+1}\right)^x \left(\frac{b-1}{W+1}\right)^y \right. \\ &\quad \cdot \left(\frac{W-i-b+1}{W+1}\right)^{n-1-x-y} \\ &\quad \left. \cdot\, p'(n-x-y, W-i-b, k-i-b) \right\}. \end{aligned} \tag{5}$$

The boundary condition for $p'(n, W, k)$ is $p'(0, W, k) = 0$ and $p'(n, 0, k) = 0$.

Using Eqs. (4) and (5), the function $p'(n, W)$ is plotted in Fig. 5 for various values of W; in this figure, the value of b is assumed to be 11. It is important to notice that the value of $p'(n, W)$ is not sensitive to W. The simulation results again match with our analysis.

4. ANALYSIS OF ASYNCHRONISM

Clock synchronization is important for power management in both DSSS and FHSS as well as for the synchronization of hopping sequence in FHSS. If the clocks of two stations are so badly out of synchronization that either power management or FHSS can not work properly, the two stations are said to be out of synchronization. If there are pairs of stations out of synchronization in an IBSS, the network is said to be in *hazardous asynchronism*. In this section, we will show that asynchronism may occur under the IEEE 802.11 timing synchronization function (TSF). For simplicity, we will often omit the adjective "hazardous" when referring to hazardous asynchronism.

Let Δ be the maximum clock difference tolerable by power management and FHSS.

Definition 2. Two clocks are *out of synchronization* if their times are different by more than Δ. Two stations are *out of synchronization* if their clocks are out of synchronization.

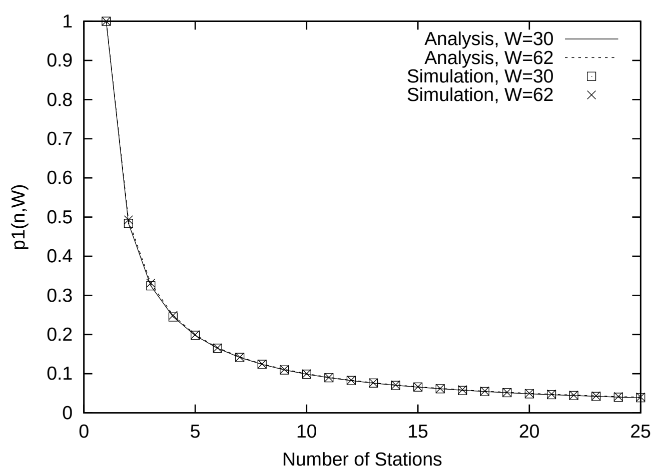

Figure 5: p'(n,W) with b=11

Assume that the clocks in an IBSS are all different in speed (or accuracy). Thus, there is a unique *fastest station*, whose clock is fastest in the system. We are particularly interested in two conditions of asynchronism:

- Fastest-station asynchronism — This refers to a situation where the clock of the fastest station is ahead of the clocks of all other stations by more than Δ units of time. Fastest-station asynchronism may occur in an IBSS because under the IEEE 802.11 timing synchronization function, stations can only set their timers forward and never backward. Slower clocks synchronize with faster clocks, but faster clocks do not synchronize themselves with slower clocks. Thus, if the fastest station fails to transmit beacons for too many beacon intervals, its clock will be ahead of all other clocks by more than Δ.
- Global asynchronism — Given a value k between one and one hundred, k percent global asynchronism (or simply k percent asynchronism) refers to the situation that at least k percent of the $n(n-1)/2$ pairs of stations are out of synchronization.

For each of these two types of asynchronism, we are interested in the following questions:

- How often may asynchronism occur in an IBSS?
- Once asynchronism occurs, how long does it last?

4.1 Fastest-Station Asynchronism

Let T be the length of a beacon interval, and let d denote the difference in clock accuracy between the fastest station and the second fastest station. Once the stations have been synchronized, if the fastest node fails to transmit a beacon in each of $\lceil \frac{\Delta}{d \cdot T} \rceil$ consecutive beacon intervals, then the fastest node will be out of synchronization with all other stations by at least Δ time units, resulting in a condition of fastest-node asynchronism. We will calculate $E(H)$, the expected lasting time of an occurrence of fastest-node asynchronism and $E(L)$, the expected length of time between two consecutive incidents of fastest-node asynchronism. H and L are illustrated in Figure 6, where the shadowed ranges indicate asynchronous periods.

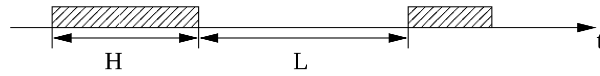

Figure 6: H and L

Given n, W, and $\tau = \lceil \frac{\Delta}{d \cdot T} \rceil$, let $p = p'(n, W)$ as in Eq. (4). Once an incident of fastest-node asynchronism occurs, it lasts at least for one beacon interval. After one interval, with probability $(1-p)^k$ the condition of asynchronism continues to hold for at least k additional beacon intervals. Thus, the expected lasting time of fastest-node asynchronism, in terms of number of beacon intervals, is:

$$E(H) = 1 + (1-p) + (1-p)^2 + \cdots = \frac{1}{p} \qquad (6)$$

Now, suppose that the fastest station has just successfully transmitted a beacon and ended a period of fastest-node asynchronism, but after exactly L beacon intervals, fastest-node asynchronism appears again. The expected value of L, $E(L)$, is an indication of how soon fastest asynchronism may occur after all clocks have been synchronized by the fastest station. $E(L)$ is the *expected inter-fastest-station-asynchronism time*. Let e_i denote the event that after a clock synchronization by the fastest station the next clock synchronization by the fastest station occurs in the ith beacon interval. Then,

$$E(L) = \sum_{i=1}^{\infty} \{\text{Prob}(e_i) \cdot E(L|e_i)\}$$

where $\text{Prob}(e_i) = p(1-p)^{i-1}$ and $E(L|e_i)$, the expected length of L given event e_i, is as follows:

$$E(L|e_i) = \begin{cases} E(L) + i & \text{if } 1 \le i \le \tau \\ \tau & \text{if } i > \tau \end{cases}$$

Therefore,

$$E(L) = \sum_{i=1}^{\tau} p(1-p)^{i-1}(E(L) + i) + (1-p)^{\tau}\tau$$

Solving this equation yields

$$E(L) = \frac{1}{p}\left(\frac{1}{(1-p)^{\tau}} - 1\right) \qquad (7)$$

Define the *fastest-node asynchronism time ratio*, denoted by R, to be the percentage of time in which the network is in fastest-node asynchronism. From Eqs. (6) and (7), it follows that

$$E(R) = \frac{E(H)}{E(H) + E(L)} = (1-p)^{\tau} \qquad (8)$$

We have derived a formula for $E(H)$, $E(L)$ and $E(R)$ for the case of fastest-node asynchronism. Using these formulae, we will compute in a later section $E(H)$, $E(L)$ and $E(R)$ for various values of n and W.

4.2 Global Asynchronism

Global asynchronism is much harder to analyze. Given a value k between zero and one hundred, let H be the duration of a continuous k percent asynchronism period, and let L be the length between two consecutive periods of k percent asynchronism. We are interested in the expected length of H and that of L.

Two clocks may get out of synchronization because their accuracies are different. Let d_k be the value such that k percent of the $n(n-1)/2$ pairs of stations differ by d_k or more in clock accuracy; in other words, $d_k = \min\{d$: at least k percent of the pairs of clocks differ by d in their accuracy$\}$. Define *d-asynchronism* as a situation where every two stations whose clocks differ in accuracy by d or more with each other are out of synchronization. Let H' be the duration of a d_k-asynchronism period, and let L' be the length between two consecutive periods of d_k-asynchronism. For simplicity, we write d_k as d in the rest of this section.

Apparently, if the IBSS is in d-asynchronism then it is in k percent asynchronism. Therefore, $E(H) \geq E(H')$ and $E(L) \leq E(L')$. If we know how to compute $E(H')$ and $E(L')$, then we have a lower bound on $E(H)$ and an upper bound on $E(L)$. Since our purpose here is to point out potential global asynchronism problems of the IEEE 802.11 timing synchronization function, a lower bound on $E(H)$ and an upper bound on $E(L)$ will serve the purpose.

Unfortunately, $E(H')$ and $E(L')$ are still difficult to compute. So, we instead compute a lower bound on $E(H')$ and an upper bound on $E(L')$, which will still serve our purpose.

As in the analysis of fastest-station asynchronism, let n be the number of stations; W, the beacon contention window size; and T, the length of each beacon interval. Observe that after the stations have been synchronized, if no node succeeds in beacon transmission in each of $\tau = \lceil \frac{\Delta}{d \cdot T} \rceil$ consecutive beacon intervals, then the condition of d-asynchronism holds. Once the system is in d-asynchronism, if no node successfully transmits a beacon in the next beacon interval, the d-asynchronous condition will last through that period. If some node succeeds in transmitting a beacon, the d-asynchronism may or may not end (because the beacon synchronizes only those clocks which are slower than the sending station). Thus, replacing the p in Eq. (6) by $p = p(n, W)$ as in Eq. (2), we obtain a lower bound on $E(H')$ and $E(H)$:

$$E(H) \geq E(H') \geq 1 + (1-p) + (1-p)^2 + \cdots = \frac{1}{p} \qquad (9)$$

Similarly, replacing the p in Eq. (7) by $p = p(n, W)$ as in Eq. (2) yields an upper bound on $E(L)$ and $E(L')$:

$$E(L) \leq E(L') \leq \frac{1}{p}\left(\frac{1}{(1-p)^{\tau}} - 1\right) \qquad (10)$$

Define the *global asynchronism time ratio*, denoted by R, to be the percentage of time in which the system is in global asynchronism. From Eqs.(7) and (6), it follows that

$$E(R) = \frac{E(H)}{E(H) + E(L)} \approx (1-p)^{\tau} \qquad (11)$$

It is not clear from Eq. 11 whether $(1-p)^{\tau}$ is a lower bound or an upper bound or neither. Our simulation results indicate that it is a lower bound. Thus, Eq. 11 is useful for demonstrating the global asynchronism problem.

5. NUMERICAL DATA ON CLOCK ASYNCHRONISM

In this section, the expected length an asynchronism interval, the expected length of time between two consecutive asynchronism intervals, and the expected asynchronism time ratio are computed using the formulae obtained in Sections 3 and 4. These data are compared with simulation results to justify various assumptions made in our analysis model. Both analytical and simulation results demonstrate the poor scalability of the current IEEE 802.11 TSF procedure.

Parameter	Value
W	30 or 62
Beacon length	11 BPs
Propagation delay	$1\mu s$
Transmission error rate	1%
aBeaconPeriod	0.1s
Clock accuracy	±0.01%
Δ	$224\mu s$

Table 2: Simulation Setup

5.1 Simulation Setup

To verify our analysis model and demonstrate the poor scalability of the IEEE 802.11 TSF procedure, we developed simulation programs in C. The simulator closely follows the protocol details of beacon generation and contention. The parameters are specified in Tables 1 and 2. Transmission errors, which were ignored in the analysis, are taken into account in simulations. Transmission errors are assumed to occur independently at destinations with an error rate of 1%. Thus, a beacon may be received with errors by some stations but without errors by others. The value of aBeaconPeriod (the length of a beacon interval) is set to 0.1s as recommended by the IEEE 802.11 specifications.

According to the IEEE 802.11 specifications, the time allocated for the PHY to hop from one frequency to another frequency in FHSS is 224μs. This value gives an upper bound on Δ, the maximum tolerable clock drift; for, if the clocks of two stations are 224μs apart, the faster station may have hopped to the new frequency and started transmitting while the slower station is still listening to the old frequency and, therefore, missing the frame. Since the 802.11-recommended value for aCurrentDwellTime is 19ms (i.e., one frequency hop per 19ms, or 52.5 hops per second) and each beacon interval lasts for 0.1s, once two stations are out of synchronization, they may miss each other's frames for 5 times before their clocks have a chance to be resynchronized in the next beacon interval. Therefore, 224μs is a generous upper bound on Δ, and it is rather conservative to let $\Delta = 224\mu s$ in our simulation and analysis.

Finally, unless otherwise indicated, each data point generated by simulation is obtained from 10 simulation runs, each run lasting for 1 hour of simulated time (or, equivalently, 36000 beacon intervals).

5.2 Global Asynchronism

The IEEE 802.11 specifications require clock accuracy to be within ±0.01%. In the worst case, two clocks satisfying the 802.11 requirement may differ by 0.02%. Assume that the values of clock accuracy of the n stations are uniformly distributed over the range $[-0.01\%, 0.01\%]$. Out of the $n(n-1)/2$ pairs of stations, at least $n^2/8$ pairs, or more than 25 percent, are different by at least $d = 0.01\%$ in their clock accuracy. Thus, a condition of d-global asynchronism with $d = 0.01\%$ will result in a condition of 25% global asynchronism, in which 25 percent of links may experience frame losses due to asynchronism. This level of asynchronism, we believe, is not acceptable by any standard. In this section, we

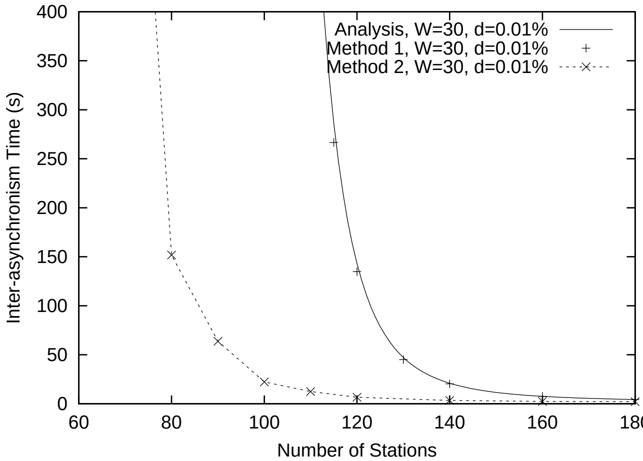

Figure 7: Inter-global-asynchronism time of IEEE 802.11

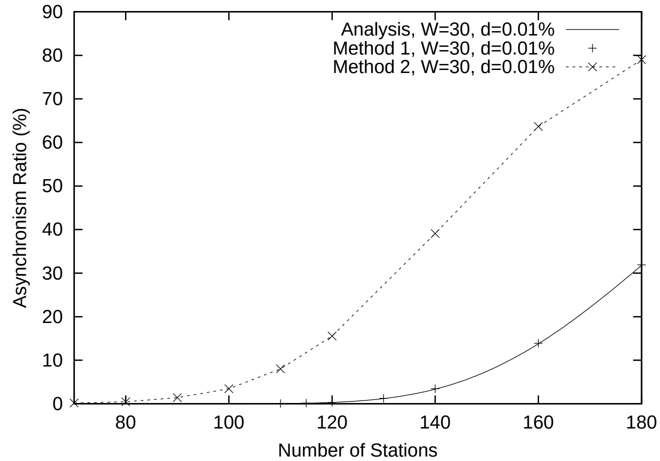

Figure 8: Asynchronism time ratio of IEEE 802.11

show by both analysis and simulation that global asynchronism is not unusual in an FHSS system.

For the length of a beacon interval, $T = 0.1$s, two clocks with a difference of d in accuracy will drift away from each other by $d \cdot T = 0.01\% \cdot 0.1s = 10\mu$s. Thus, with $\Delta = 224\mu$s, $d = 0.01\%$ and $T = 0.1$s, we have $\tau = \lceil \frac{\Delta}{d \cdot T} \rceil = 23$. Using Eqs. (2), (3), and (7), we compute a theoretical upper bound on the expected inter-d-global-asynchronism time, $E(L) \cdot T$, for $W = 30$ and various values of n. The results are plotted as a solid curve in Fig 7. From the figure, it is observed that when $n = 110$, the IBSS suffers from d-global asynchronism and thus from 25% global asynchronism, once every 7.5 minutes (450s) on average. The situation gets worse rapidly as the number of stations increases. For instance, when $n = 150$, 25 percent asynchronism occurs every 10.4 seconds. And these are only upper bounds. The actual frequency of asynchronism could be much worse.

In our simulation, the inter-asynchronism time is measured in two ways. First, we follow exactly the approach used in the analysis and count the number of consecutive beacon intervals during which no beacons are successfully sent. When this number exceeds $\tau = 23$, we record an event of asynchronism. In the second way, we count the number of pairs of stations which are out of synchronization. Whenever there are more than 25 percent of pairs out of synchronization, we record an incident of global asynchronism (so we are observing 25 percent global asynchronism).

Figure 7 shows the simulation results by both methods. As expected, the data points generated by method 1 fit the theoretical line very well. The data points on the dotted line are obtained by method 2. From these data, we observe that 25 percent asynchronism occurs more often than as predicated by analysis. (This not surprising since the analysis gives only an upper bound of the expected inter-global-asynchronism time.) A network of 80 stations suffers from an attack of 25 percent global asynchronism once every 2.5 minutes. For a network of 100 stations, the attack takes place every 20 seconds.

Figure 8 shows the global asynchronism time ratio. Once again the results of simulation with method 1 fit the theoretical upper bound precisely while the results generated with method 2, which measures the 25 percent asynchronism time ratio, indicate that the actual asynchronism problem is more sever than as predicated by the upper bound. For example, for an IBSS of size 160, at least 25 percent of the station pairs are out of synchronization for about 65% of the time. Such a ratio is undesirably high for communication in an IBSS.

The IEEE 802.11 specifications define the beacon contention window, W, to be 30 for FHSS systems. For such systems, the above results suggest that the current 802.11 DSF procedure does not scale very well. As for DSSS, where $W = 62$, the situation is different. Our analysis shows that even for $n = 200$ the global asynchronism is expected to occur only after 148 days. Thus, global asynchronism is not a problem for a DSSS system due to its larger beacon contention window size W. However, as to be seen in the next section, fastest-station asynchronism is still a problem for large-scale DSSS networks.

5.3 Fastest-Node Asynchronism

Assume that the clock of the fastest station is faster in speed (or accuracy) than the clock of the second fastest station by d, where $d = 0.005\%$ or $d = 0.003\%$. We compute the expected time between two consecutive incidents of fastest-node asynchronism for different values of d and W using Eqs (4) and (7). The results are shown in Figure 9.

In the simulation, we let the fastest clock and the second fastest clock differ in accuracy by d, while the other clocks are uniformly distributed in the range of $[-0.01\%, \rho_2]$, where ρ_2 is the accuracy of the second fastest clock. We compare the timing value of the fastest station with that of other stations. Whenever the fastest node is out of synchronization with all other stations, we record an incident of asynchronism. Once again our analysis results fit with the simulation results closely. From the figure, we see that asynchronism happens even when n is relatively small. In this sense, fastest-node asynchronism is much more likely to occur than global asynchronism. Furthermore, a larger window size ($W = 62$) does not perform any better than a smaller window size ($W = 30$). This observation is consistent with an earlier remark that $p'(n, W)$ is insensitive to the value of W.

The fastest station is more likely to experience asynchronism than other stations because according to the TSF procedure clocks move only forward and never backward. Fastest-

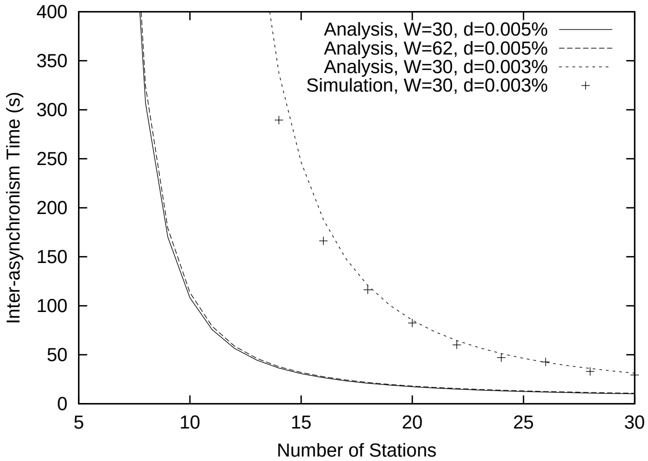

Figure 9: Inter-fastest-node-asynchronism time of IEEE 802.11

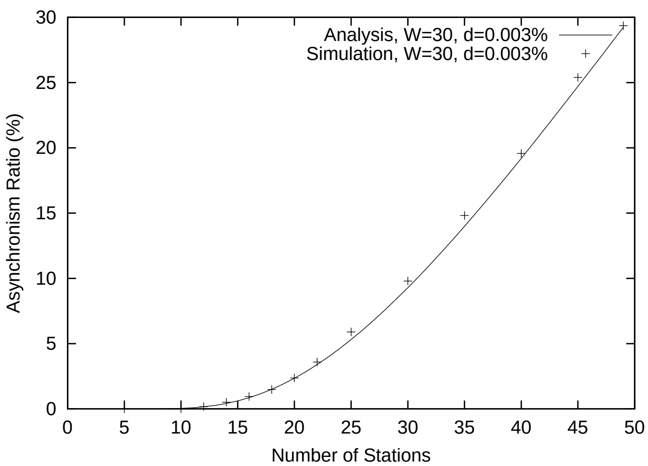

Figure 10: Fastest-node asynchronism time ratio of IEEE 802.11

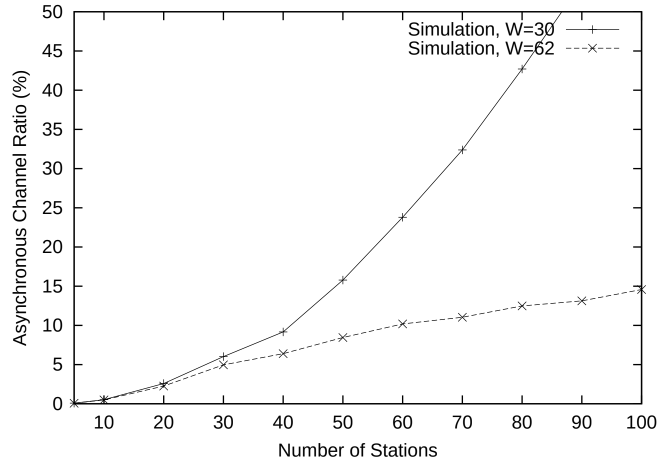

Figure 11: Average fastest-node asynchronous channel ratio of IEEE 802.11

node asynchronism occurs when the fastest station can not successfully transmit beacon frames for a number of consecutive beacon periods. Since this probability is much larger than the probability that no station at all successfully sends a beacon frame for the same number of beacon intervals, fastest-node asynchronism is more likely to occur than global asynchronism. This phenomenon can be observed by comparing Figure 7 to Figure 9.

The fastest-node asynchronism time ratio when $W = 30$ is given in Figure 10, from which we observe that the fastest station is out of synchronization with all other stations for 10% of the time even for an IBSS of just 30 stations. With a network of 50 stations, it is under fastest station asynchronism for 30% of the time.

Next, we examine fastest-node asynchronism by simulation, assuming that all the clocks are uniformly distributed in accuracy between -0.01% and $+0.01\%$. This time we are interested in the number of stations with which the fastest station is out of synchronization. The results are shown in Fig. 11, where the y-axis is the average percentage of stations with which the fastest station is out of synchronization. The problem of asynchronism is more severe for FHSS ($W = 30$) than for DSSS ($W = 62$). In an FHSS system of 40 stations, the fastest station is, at any time instance, out of synchronization with, on average, 10% of the stations. When the IBSS size is 80, the fastest station would have hard time communicating with 45% of the stations. For $n = 100$, the fastest station would have synchronization problems with almost every other station. DSSS systems are not immune to the problem. For $n = 100$, the fastest station is, on average, out of synchronization with 15% of stations at any moment. The fastest-node asynchronism is a real problem for DSSS as well as for FHSS.

6. A SCALABLE CLOCK SYNCHRONIZATION PROCEDURE

We have seen clock asynchronism problems that may arise when scaling up an IBSS. Now let us turn to the question of fixing it.

If stations were allowed to set their timers both forward and backward in the process of synchronization, then there would be no fastest-node asynchronism but just global asynchronism problem, which could be alleviated by enlarging the beacon contention window size W. A value of 62 for W, as used in DSSS, could practically allow an IBSS of size 200. If a fixed large window size is not desirable, a simple scheme allowing the window size to dynamically change has been observed to perform very well. This result is not reported in this paper due to space limit.

Now, suppose we must keep intact the IEEE 802.11 TSF's provision of moving forward only. Since the stations can only set their timers forward, why not give the faster stations a higher priority to send beacon frames? Based on this idea, we propose an *adaptive timing synchronization procedure* (ATSP) that can considerably alleviate both the problem of fastest-node asynchronism and that of global asynchronism. We assign to each station i an integer $I(i)$ that determines how often each station shall participate in beacon contention. That is, station i contends for beacon transmission once every $I(i)$ beacon periods. Therefore, smaller the value of $I(i)$, higher the station's chance of beacon transmission. Let the maximum possible value of I be $I_{\max}$. Let $C(i)$ be a counter at station i that counts beacon periods. The proposed clock synchronization algorithm is described in the following. It is a simple modification to the current IEEE 802.11 TSF, and can be easily implemented.

ATSP: Adaptive Timing Synchronization Procedure

1. Initially, for each station i, let $I(i)$ be a random number between 1 and $I_{\max}$ and let $C(i) := 1$.

2. In each beacon interval, station i participates in beacon contention iff $C(i) \mod I(i) = 0$.

3. Whenever station i receives a beacon with a timing value later than its own, the station sets its timer to this value, increases $I(i)$ by 1 if $I(i) < I_{\max}$, and sets $C(i) := 0$.

4. If station i does not receive any beacon frame with a timing value later than its own for I_{max} consecutive beacon intervals, it decrements $I(i)$ by 1 if $I(i) \geq 2$, and sets $C(i) := 0$.

5. At the end of a beacon interval, each station increases its $C(i)$ by 1.

With ATSP, once the clocks have been synchronized, the fastest station will not receive a timing value greater than its own; its I-value will gradually decrease to 1 and stay there. The other stations will gradually increase their I-values until they reach $I_{\max}$. Define a *stable state* to be one in which the fastest station's I-value has reached 1 and the other stations have reached I_{max}. Once the IBSS reaches a stable state, the fastest station has a very high probability of successfully sending a beacon and thereby synchronizing all the other stations. In the above analysis, we implicitly assume, for ease of presentation, that there is only one fastest station. This assumption is not essential. If there are multiple fastest stations, ATSP will still works well.

To testify the performance of ATSP, we did a couple of simulations assuming a static IBSS adopting the parameter used in Section 5. For global asynchronism, we adopted d=0.01% and τ=23; and chose 10 for $I_{\max}$. For different values of $n \leq 400$, we ran simulation for 20 times, each for 20 minutes of simulated time. Not a single incident of global asynchronism was observed. We also ran simulation for fastest-station asynchronism, with $d = 0.003\%$ and with completely randomized clocks, and once again observed not a single case of asynchronism.

Then we evaluated the performance of ATSP for a dynamic IBSS. In ATSP the fastest station is, to some extent, in charge of timing synchronization. Should the fastest station leave the IBSS because of mobility or other reasons, the formerly second fastest node will become the fastest. Suppose its index is j. Since $I(j)$ is decreased by one in every $I_{\max}$ beacon intervals, it takes station j at most $I_{\max}^2$ beacon intervals to reduce $I(j)$ to 1. For example, if $I_{\max} = 10$, the time will be at most 100×0.1s = 10 seconds. Note that restabilization does not mean asynchronism; during restabilization, there may or may not be any incidence of asynchronism. In order to understand the probability of asynchronism in the process of restabilization, we let the fastest station leave and return to the IBSS once every six minutes. That is, it alternately leaves the IBSS for three minutes and returns for the next three minutes.

As expected, no asynchronism was caused by a fastest node joining the IBSS. On the other hand, in the restabilization process caused by the leaving of the fastest station, asynchronism did occur but with a very low probability. The results are shown in Table 3. For $n = 400$ and $d = 0.01\%$, eight cases of global asynchronism happened in 1000 rounds of restabilization and each time the IBSS suffered from asynchronism for only 3.3 seconds on average. For $n = 400$ and $d = 0.003\%$, fastest-station asynchronism with $d = 0.003\%$ occurred for seven times out of 1000 rounds of restabilization and the average asynchronism time was 1.2 seconds. When $n = 300$ no asynchronism was observed. For fastest-node asynchronism with completely randomized clocks, not a single occurrence of asynchronism was observed. Due to the limited bandwidth of IEEE 802.11 ad hoc networks and the limit of MAC protocol, 300∼400 is a large size for an IBSS. Therefore, the restabilization process is practically safe from asynchronism.

n, number of stations	300	350	400
Global asyn., d=0.01%	0	1	8
Fastest-station asyn., d=0.003%	0	0	7
Fastest-station asyn., randomized	0	0	0

Table 3: Occurring times of asynchronism in 1000 rounds of restabilization

In summary, the adaptive timing synchronization procedure greatly alleviates both global asynchronism and fastest-node asynchronism; it shows stable performance in responding to mobile stations.

7. CONCLUDING REMARKS

We have pointed out the clock asynchronism problem faced by a large-scale IBSS, and have proposed a simple scheme to fix it. Our studies were based on the IEEE Std 802.11, 1999 Edition. An interesting question is whether the results reported here also apply to 802.11b LANs as specified in [5]. The answer is affirmative. First, clock synchronization is still an important requirement since the PHY layer of 802.11b still uses both DSSS and FHSS with power management. Second, our analyses in Sections 3 and 4 still hold for 802.11b as the analyses did not assume any particular values for Δ, T, W, b, and τ. The numerical and simulation results of Section 5 were obtained using the parameter values of Std 802.11 and would not necessarily apply to 802.11b. For 802.11b, the value of Δ remains the same, 224μs; T remains the same, 0.1s; d remains the same since the clock accuracy requirement does not change; $W = 62$, bigger than the value 30 used in our simulations; b becomes smaller due to the higher data rate and hence shorter beacon transmission time. In this case, the IBSS can support up to about 250 stations without causing many incidences of global asynchronism. However, our simulation showed that when $n \geq 300$ global asynchronism becomes a severe problem again. As for fastest-node asynchronism, we have seen that this problem cannot be solved by just increasing the beacon contention window size. Thus, fastest-node asynchronism is still a big problem for IEEE 802.11b despite its larger window size, $W = 62$. The exciting thing is that our ATSP protocol will work even better for IEEE 802.11b than for IEEE Std 802.11 because the larger window size will relieve the beacon contention. Therefore, it would not be a surprise if ATSP can effortlessly support more than 500 stations with IEEE 802.11b.

The IEEE 802.11 IBSS is a fully connected, single-hop, ad hoc network, with all stations within each other's transmission range. An interesting question is whether 802.11 can be used to implement multi-hop ad hoc networks, assuming that each station has been enhanced with routing capability. We are currently investigating this problem and will report our results in a future paper.

8. REFERENCES

[1] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, 2000.

[2] G. Bianchi, L. Fratta, and M. Oliveri. Performance evaluation and enhancement of the CSNA/CA MAC protocol for 802.11 wireless LANs. In *Proc. PIMRC 1996*, pages 392–396.

[3] J. Elson, L Girod and D. Estrin. Fine-grained network time synchronization using reference broadcasts. Submitted for review, February 2002. http://lecs.cs.ucla.edu/Publications/Publications.html.

[4] IEEE Std 802.11. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification*, 1999 edition edition.

[5] IEEE Std 802.11b. *Higher-Speed Physical Layer Extension in the 2.4 GHz Band*, 1999 edition edition.

[6] L. Lamport. Time, clocks and the ordering of events in distributed systems. *Communications of the ACM*, 21(7):558–565, 1978.

[7] J. Li, C. Blake, D. S. J. D. Couto, H. I. Lee, and R. Morris. Capacity of ad hoc wireless networks. In *Proc. 7th ACM International Conference on Mobile Computing and Networking*, 2001.

[8] D. L. Mills. Internet time synchronization: the network time protocol. In *Global States and Time in Distributed Systems*. IEEE Computer Society Press, 1994.

[9] K. Römer and E. Zurich. Time synchronization in ad hoc networks. In *Proceedings of MobiHoc 2001*.

[10] Y. C. Tay and K. C. Chua. A capacity analysis for the IEEE 802.11 MAC protocol. *ACM/Baltzer Wireless Networks*, 7(2):159–171, 2001.