A Measurement Study of Path Capacity in 802.11b based Wireless Networks

Tony Sun, Guang Yang, Ling-Jyh Chen, M.Y. Sanadidi, Mario Gerla
Department of Computer Science, UCLA
{tonysun, yangg, cclljj, medy, gerla}@cs.ucla.edu

Abstract

The rapid deployment of wireless networks in various environments necessitates the development of new end-to-end tools that monitor and measure the properties of wireless paths well. In this paper, we implement AdHoc Probe, a recently proposed path capacity estimation tool specially designed for the multi-hop ad hoc wireless environment. We present an implementation of AdHoc Probe in Linux; discuss challenges in its deployment, and solutions to the various system issues encountered. We also evaluate the behavior/effectiveness of AdHoc Probe in various testbed setups; including an interfered setting that cannot be simulated. Experiment results validate the workings of AdHoc Probe and offer insights into how the capacity of a wireless path changes in real wireless environments. Our efforts provide a basis for realistic results that can be of assistance in activities such as capacity planning, protocol design, performance analysis, and etc.

1. Introduction

As the wireless Internet grows in size and connection complexity, new applications and services requirements necessitate the development of suitable end-to-end tools that monitor and measure the properties of wireless paths well. In particular, effective evaluation and measurement of the capacities along wireless paths are of realistic interest, especially to activities such as capacity planning, protocol design, performance analysis and system deployment.

Although the path capacity estimation problem has been extensively studied in the literature, most tools work in the scenarios of wired and/or last-hop wireless networks and measure the bottleneck link capacity, e.g. [1][5][8][10]. The complexity and convergence time required for these schemes are not well suited for multi-hop ad hoc wireless networks. Moreover, the assumption of bidirectional setup of some of the above techniques has proved to yield detrimental results in ad hoc networks.

In fact, end-to-end path capacity estimation in ad hoc wireless networks is much more challenging. Wireless capacity estimation depends not only on the rate of the "narrow" link along the path (as in a wired network), but also on the topology, path layout, interference between nodes along the path and several other environmental parameters. Moreover, wireless capacity estimation must accommodate both fixed rate wireless networks, and auto rate wireless networks, where the transmission rate can be dynamically adjusted to the propagation characteristics and energy requirements. An accurate capacity estimation mechanism must understand how the capacity of a wireless path is impacted by these factors respectively.

Previous technique proposed by Li et al in [11] addressed the adhoc end-to-end path capacity by sending a brute force UDP packet stream to measure the maximum achievable throughput. This scheme produces very realistic results, but is not very practical since it heavily impacts existing traffic, and its result is affected by current on-going traffic conditions as well.

AdHoc Probe, a recently proposed adhoc path capacity estimation tool by Chen et al [1], is a simple and effective technique that aimed to simplify the adhoc path capacity estimation process. However, performance of AdHoc Probe was only studied through NS2 [19] simulations in [1]. Little understanding exists on the challenges in deploying AdHoc Probe, as well as the effectiveness and behavior of AdHoc probe in real wireless environments. Therefore, it is the purpose of this study to reveal how AdHoc Probe would perform in real wireless environments.

Motivated by the above goals, we implement AdHoc Probe in Linux and conduct in-depth evaluations in various controlled wireless testbed, to assess the efficacy of AdHoc Probe in a real network deployment. Important system issues commonly associated with such end-to-end measurement tools, such as time synchronization and clock skew, are discussed with proposed solutions. We also evaluate the effect of auto rate wire-

less radios experimentally by varying the distances between nodes as well as interfered settings/patterns that cannot be simulated via NS2. An important differentiation between auto rate radio and multi-hopping, in terms of the impact on wireless path capacities, is also among our goals.

Our experiment results show that AdHoc Probe is able to accurately measure the wireless path capacity in all cases of fixed rate networks. Moreover, AdHoc Probe is able to track the dynamic rate adaptation of an auto rate wireless link in a timely and accurate manner.

The rest of the paper is organized as follows. In section 2, we briefly describe the AdHoc Probe mechanism. An in-depth discussion of related system issues and their solutions is presented in Section 3. In section 4, we present testbed experiment results to evaluate the efficacy of AdHoc Probe in both fixed and auto rate wireless networks. Finally, section 5 concludes the paper.

2. AdHoc Probe Overview

AdHoc Probe is an end-to-end tool that measures the capacity of a multi-hop wireless path in an ad hoc network. Inspired by CapProbe [8], AdHoc Probe relies on the packet-pair technique to provide capacity estimation in wireless networks. However, while CapProbe is designed to estimate the *bottleneck link* capacity in a round-trip fashion, AdHoc Probe intends to estimate the *end-to-end path* rate based on *one-way* measurements. The end-to-end path rate is the *maximum achievable rate* over the wireless path in the absence of any competing traffic. The maximum *achievable rate* (in the absence of competing traffic) is typically lower than the nominal channel transmission rate due to a variety of reasons including multi-hopping, unique features of wireless networks (e.g. RTS/CTS mechanism), link rate adaptation techniques, etc. AdHoc Probe is able to accurately measure such achievable rate.

The basic one-way AdHoc Probe algorithm is derived from the round-trip CapProbe mechanism. Probing packet pairs of fixed size are sent back-to-back from the sender to the receiver. The sending time is stamped on every packet by the sender, the One Way Delay (OWD) is calculated at the receiver, and the path capacity (i.e. rate) estimation is performed at the receiver and communicated to the sender.

The receiver measures the OWD of each packet as the difference between the receiving time (clocked at the receiver) and the sending time (stamped by the sender in the packet header). The OWD sum is then computed. The "good" packet-pair samples (i.e. the packet pairs encountering no cross traffic) are those with minimum OWD sum (as shown in Figure 1), and the corresponding capacity is given by $C=P/T$, where P is the packet size and T is the dispersion of the packet pair.

AdHoc Probe does not implement the "convergence test" as CapProbe does in order to make the algorithm simple, fast, and timely to the highly varying characteristics of wireless networks. Instead, AdHoc Probe simply reports the capacity estimation after receiving a certain number of samples, say N. Similar to CapProbe, the accuracy of capacity estimation increases as N grows. However, a large N value is not suitable for mobile wireless networks as it will lead to high latency in estimation and may not allow us to capture the dynamic properties of the wireless network.
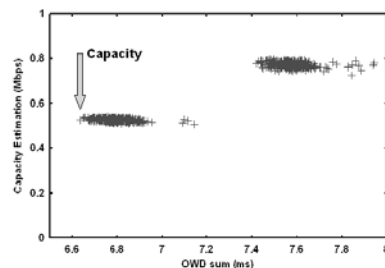

Figure 1: AdHoc Probe capacity estimate using the sample with minimum OWD sum.

Apart from the number of samples, N, the latency of the estimate also depends on the probing packets sending rate (i.e. the probing rate). For simplicity, AdHoc Probe simply sends probing packets (with the packet size of P bytes) using a CBR rate (i.e. R packet-pair/second, or equivalently $2*P*R$ bytes/second). As a result, the expected duration for one estimate is approximately N/R seconds. Clearly, the larger R is, the less time a capacity estimation process takes. However, R should be upper-bounded since a large R may disturb the ongoing foreground traffic in the network or even congest the network. As a result, the capacity estimate may become inaccurate (hard to get one good sample) or extremely slow (packets are lost due to congestion).

The probing parameters N and R need to be carefully tuned in accordance with the underlying network properties and by trading off precision for speed. This trade-off clearly depends on the application. In this paper, we simply set $N = 200$, $P= 1500$, and $R = 4$ sample pairs/second for the testbed experiments.

3. System Issues

In this section, we will discuss the seriousness of system issues encountered and present solutions that mitigate those problems. In subsection 3.1, we will first discuss existing Auto Rate schemes in wireless radios, allowing us to properly investigate and distinguish how Auto Rate radios and wireless multi-hopping affects wireless capacity, respectively. We will then discuss the clock screw problem in subsection 3.2. Finally, we describe the system time synchronization issue and report the implementation details of necessary "correction" in subsection 3.3.

3.1. Wireless Auto Rate

In our experiments, it is important to be aware of how existing Auto Rate schemes work, so we can properly investigate how Auto Rate radios influence wireless capacity. Auto Rate functionality is important for multi-rate wireless devices to maximize the utilization of network resources. For instance, by simply adjusting the transmission data rate, one can achieve higher data throughput with the higher data rate mode, or increase the transmission range and robustness against interference by using a lower data rate mode. Additionally, even within the same data rate mode, the overall data throughput can be improved by opportunistically taking advantage of the channel coherence time (duration for which the wireless node has better-than-average channel conditions). Finally, data rate can be changed to save energy [15].

Several auto rate schemes have been proposed to exploit the multi-rate capability. They can be categorized into two types: *Adaptive Rate* schemes (e.g. ARF [7], RBAR [4], and AARF [9]) and *Opportunistic Scheduling* schemes (e.g. OAR [16] [18] and MAD [6]).

Adaptive Rate schemes could be either sender based or receiver based. Auto Rate Fallback (ARF) [7] is the first published and implemented sender based rate adaptation algorithm. The basic idea of ARF is to start the transmission using highest rate and switch to lower rate when 1 or 2 consecutive failures occur. ARF also starts a timer upon rate dropping. When either the timer expires or 10 successfully received acknowledgements are counted, the transmission rate is upgraded to a higher rate and the timer is reset. The drawbacks of ARF are: (a) the heuristic based ARF cannot adapt effectively in a rapidly varying wireless channel, and (b) ARF data rate tends to suffer from high oscillation even when the wireless channel is not rapidly changing. In [9], Mathieu et al propose Adaptive ARF (AARF) to adapt

the threshold settings of ARF in accordance to the channel conditions. As a result, the frequent rate oscillations in ARF are mitigated.

3.2. System Time Synchronization Issue

The OWD measurement in AdHoc Probe is indeed problematic on a real testbed. Unlike the perfect time synchronization provided by the simulator, the system clocks of the two end hosts are usually not synchronized. As a result, the measured OWD will not be identical to the actual OWD. Though some software packages and service protocols (e.g. NTP [12]) have been proposed to enable time synchronization of network hosts, one can not guarantee the two end hosts are always synchronized before the estimation. Thus, a successful capacity estimation technique should not rely on any assumptions of a perfectly time-synchronized system. We now provide simple analysis on the AdHoc Probe algorithm and show that AdHoc Probe works well even when the system time is not perfectly synchronized.

Suppose δ is the constant time offset between the AdHoc Probe sender and receiver. For the i-th packet pair sample, the sending time is stamped $T_{send,i}$, and the receiving times (on the receiver) are $T_{recv1,i}$ for the first packet and $T_{recv2,i}$ for the second packet, respectively. Therefore, the measured OWD sum (S') and the actual OWD sum (S) of the i-th packet pair sample are:

$$S_i' = (T_{recv1,i} - T_{send,i}) + (T_{recv2,i} - T_{send,i}) \tag{1}$$

$$\begin{aligned} S_i &= (T_{recv1,i} - T_{send,i} - \delta) + (T_{recv2,i} - T_{send,i} - \delta) \\ &= S_i' - 2\delta \end{aligned} \tag{2}$$

Thus the difference between S_i and S_i' is a constant 2δ for all packet pairs. If $S_k = \min(S_i)$ for $i = 1 \ldots n$, $k \in [1 \ldots n]$, then $S_k' = \min(S_i')$ for $i = 1 \ldots n$ and vice versa. By filtering out those samples of non-minimum S', it is easy to identify the "good sample" that has the minimum S' and S, and the capacity estimation is computed by using the interval between this packet pair sample. Clearly, the interval is not affected by the time offset. Therefore, AdHoc Probe is able to absorb the constant time offset δ between the sender and the receiver and produce an accurate capacity estimate.

3.3. Clock Screw Issue

In additional to the time synchronization issue, the deployment of one-way AdHoc Probe may also suffer from the clock skew problem, i.e. the clock "drift" is not identical on different machines. Figure 2 shows an example of actual OWD measurements, when we send

UDP packets (4 packets per second) from one laptop to the other using 802.11b. The relative OWD (i.e. OWD_i-OWD_1 for the i-th measurement) can be skewed by 1 millisecond after only 80 packets (i.e. 20 seconds)! This is a very big error relative to the typical delay sum, in the order of tens of ms (as seen in Figure 2). As a result, when the OWD measurements are affected by an increasing (decreasing) skew, early (late) sample tend to get selected by AdHoc Probe as the "good" sample.
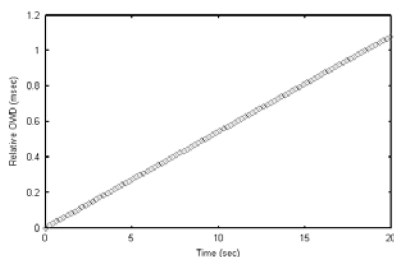

Figure 2: Illustration of clock skew problem in OWD sum measurements.

More specifically, when the clock drift is skewed, the time offset of the two machines is not constant. We use $\Delta(t)$ to denote the time offset function, where t is the system time at the receiver. The actual OWD sum of the i-th packet pair sample is then revised as

$$S_i = (T_{recv1,i} - T_{send,i} - \Delta(T_{recv1,i})) + (T_{recv2,i} - T_{send,i} - \Delta(T_{recv2,i})) \quad (3)$$
$$= S_i' - \Delta(T_{recv1,i}) - \Delta(T_{recv2,i})$$

Unlike the result in the previous subsection, $S_k'=\min(S_i')$ for $i = 1 \dots n$ does not infer $S_k=\min(S_i)$ for $i = 1 \dots n$. It turns out that the sample with the minimum measured OWD sum is not guaranteed to be the "good sample", which has the minimum actual OWD sum. Therefore, in order to obtain the "good sample", it is necessary to *calibrate* the measured OWD sum and remove the effect of $\Delta(t)$.

The calibration problem has been first studied in [14], in which Paxson used forward and backward path delay measurements to deal with this problem. However, this approach requires some heuristic tuning, and it is not feasible for pure one-way estimation. The linear regression algorithm is discussed in both [13] and [14]. However, it only works well if the network delays are normally distributed. [13] also formulates this problem as a linear program and solves it using standard algorithms in [3]. In this study, we employ a convex-hull based approach [17], which gives results similar to the linear programming approach, and in addition can handle clock resets and velocity adjustments as well. Moreover,

the complexity of the convex-hull based approach is $O(n)$, and is easily embedded in AdHoc Probe.

Here, we assume the time offset function $\Delta(t)$ is monotonic with either an increasing or a decreasing trend. The trend of $\Delta(t)$ can be determined by the *alg_trend* algorithm, as shown in Figure 3 with a manual threshold setting K. The smaller K is, the more sensitively the algorithm behaves. In this paper, we set $K=N/3$ for all experiments.

```
Algorithm alg_trend:
  trend = 0;
  For i = 2 to N
    If S'i > S'i-1 then trend++;
    If S'i < S'i-1 then trend--;
  End for
  If trend>K, Δ(t) is with increasing trend
  If trend<K, Δ(t) is with decreasing trend
```

Figure 3: Algorithm for determining the trend of offset function, $\Delta(t)$.

Once the trend of $\Delta(t)$ is found, the convex-hull calibration algorithm is applied. As shown in Figure 4, the calibration algorithm [17] will find a lower bound of skewed measurements when the trend is increasing or an upper bound when the trend is decreasing. We assume $\Delta(t)$ is a piecewise linear function; therefore, it is sufficient, but not necessary, that all good samples lie on the lower/upper bound curve.
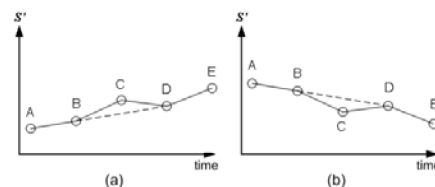

Figure 4: Illustration of convex-hull based approach. (a) the increasing trend case; (b) the decreasing trend case.

Suppose Ω_1 denotes the set of data points lying on the bound curve of OWD measurements of the first packet in the probing samples, and $\Omega_{S'}$ denotes the set of data points lying on the bound curve of S'. Ideally (i.e. each data point in Ω_1 is with the minimum OWD of the first packet, and each data point in $\Omega_{S'}$ is with the minimum OWD sum), $\Omega_{S'} \subseteq \Omega_1$. However, since Ω_1 and $\Omega_{S'}$ may also contain samples that are "not good", $\Omega_{S'} \subseteq \Omega_1$ does not always hold. In order to improve the accuracy of AdHoc Probe estimation, we identify the good samples by taking the intersection of Ω_1 and $\Omega_{S'}$, since a good sample must have both the minimum OWD of the first packet, and the minimum OWD sum. For each sample in the intersection set of Ω_1 and $\Omega_{S'}$, one capacity esti-

mation is made accordingly. AdHoc Probe then reports its end-to-end path capacity estimation result by taking the average of those estimates.

4. Testbed Experiments

Here, we perform testbed experiments to measure path capacities of wireless ad hoc networks using AdHoc Probe. Implementation issues such as time synchronization and clock skew are addressed. We experiment with AdHoc Probe in both fixed rate and auto rate actual wireless configurations in the lab. Auto rate adjustments are induced by varying the physical distances between nodes and by subjecting the 802.11b links to Bluetooth interference.

4.1. Experimental Results in Fixed Rate Wireless Networks

The testbed was first set to validate the path capacity on multi-hop fixed rate wireless networks. We placed several 802.11b laptops about 70 ~ 80 meters apart in a chain topology. The wireless rate was fixed at 2Mbps. 20 capacity estimates were collected for each path length (i.e. each number of hops). Each run included 200 packet pair samples, and 4 samples were injected every second. The experiment is conducted without cross traffic, and the average and standard deviation of the capacity estimates is presented below in Figure 5.
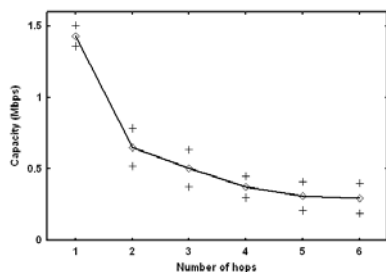

Figure 5: Experiment results of AdHoc Probe on wireless multihop testbed (transmission rate is 2Mbps on each link).

From the results, it is obvious that the effective capacity of a chain topology decreases as the hop length increases, and the estimate remains constant after the number of hops becomes larger than 4.

4.2. Experimental Results of Auto Rate Wireless Networks Triggered by Displacements

To experimentally validate the relationship between source-destination distance and path capacity, we measured the path capacity between auto rate capable nodes when the distance varies by 20 meter increments. The data transmission rate can adapt in the range {11Mbps, 5.5Mbps, 2 Mbps, 1 Mbps}. Four AdHoc Probe samples were collected every second and each run consists of 200 samples. The experiment was conducted without cross traffic. 20 capacity estimates were collected, and their average and standard deviation are presented below in Figure 6.
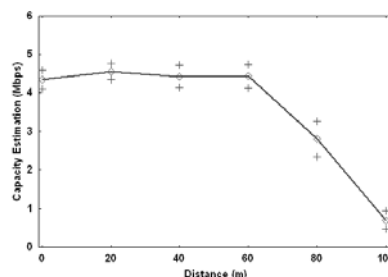

Figure 6: Experiment results of 802.11b one hop connection (auto rate) with different distance between two hosts.

From the results, the estimated capacity remains basically unchanged when the source-destination pair is within 0-60 meters (the average effective one-way capacity is approximately 4.4Mbps, which corresponding to the 11Mbps modem rate when the various O/H components are factored out). When the distance between the source and the destination node increases beyond 60 meters, we observed a decrease in the measured capacity. In particularly, when the distance between the source-destination reaches 80 meters, AdHoc Probe measures an average effective capacity of ~3Mbps, corresponding to 5.5 Mbps modem rate. When the distance between source-destination reaches 100 meters, we measured an average effective capacity of ~1Mbps, which again, corresponds to 1Mbps modem rate[1]. The experimental results thus confirm the relationship between source-destination distance and path capacity.

4.3. Experimental Results with Bluetooth Interference

Rate adaptation can be triggered not only by a change in distance but also by wireless interference. In fact, interference has the same effect as reducing the signal to noise ratio as distance does.

---

[1] The effective capacity of a one-hop 802.11b link can be found at http://www.uninett.no/wlan/throughput.html

To investigate the influence of wireless interference on effective capacity of a wireless link, we set up an experiment with a single hop 802.11b path interfered by Bluetooth. Figure 7 illustrates the testbed configuration. Two 802.11b laptops (i.e. AdHoc Probe sender and receiver) are placed 10 meters apart, and two Bluetooth laptops (the interference generators) communicate with each other creating interference to the 802.11 receiver. The Bluetooth pair is placed at a varying distance from the 802.11 receiver (from 0 to 9m). The Bluetooth source sends a CBR traffic to the Bluetooth receiver at 240kbps (1500 bytes/packet; 20 packets/second). Since Bluetooth and 802.11b use the same radio frequency band (i.e. 2.4GHz), they interfere with each other, and the link quality of the 802.11b connection degrades. As a result, the 802.11b sender adjusts its rate using ARF attempting to adapt to the changing channel conditions.
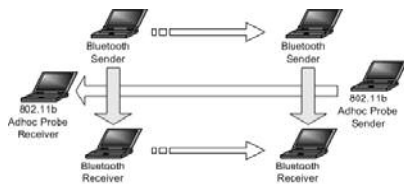

Figure 7: Auto Rate 802.11b Testbed with Bluetooth interference.

For each data point，20 AdHoc Probe tests were made, each test consisting of 200 packet pair samples. Probing rate is 5 packet-pairs per second. The average and standard deviation of the capacity estimates are presented below in Figure 8.
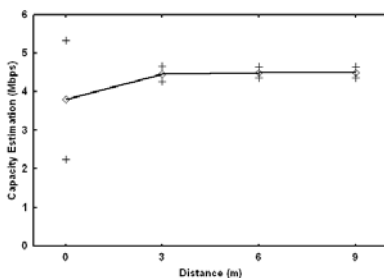

Figure 8: Experiment results of 802.11b with auto rate and with Bluetooth interference from varying distance.

From the results, the average capacity estimate is consistently in the 4Mbps range, which is what we expect for a single hop 11Mbps channel. The estimate is very sharp for Bluetooth beyond 3m. For zero distance, the estimate oscillates as the Auto Rate controller tries to keep up with the changes. It is remarkable that the average estimate at zero Bluetooth distance is quite close to the actual rate.

5. Conclusion

In this paper, we have implement and conduct an in-depth evaluation on the effectiveness of AdHoc Probe, a newly proposed end-to-end path capacity estimation tool designed for multi-hop ad hoc wireless networks. Important system issues associated with the mechanism, such as time synchronization and clock skew, were discuss in detail, and solutions were presented. Effects of the auto rate wireless radio on wireless path capacity are also studied and evaluated experimentally by varying the distances between nodes as well as the interference patterns. We have also investigated the respective effects of auto rate radio and multi-hopping on wireless path capacities. Our experiment results have shown that AdHoc Probe is a useful and practical tool that can indeed be deployed in real wireless networks. It is able to accurately measure the wireless path capacity in all cases of fixed rate networks, and track the rate adaptation of an auto rate wireless link timely and correctly.

Acknowledgement

References

[1] Chen, L.-J., Sun, T., Yang, G., Sanadidi, M. Gerla, M., *"AdHoc Probe: Path Capacity Probing in Ad Hoc Networks"* UCLA Computer Science Technical Report TR050005.

[2] Dovrolis, C., Ramanathan, P., and Moore, D., *"What do packet dispersion techniques measure?"* IEEE Infocom'01.

[3] Dyer, M. E., "Linear Algorithms for Two- and Three- variable Linear Programs," SIAM Journal on Computing, 13 (1983), 31-45.

[4] Holland, G., Vaidya, N., and Bahl, P., "A rate-adaptive MAC protocol for multi-hop wireless networks," MobiCom 2001.

[5] Jacobson, V., *"Pathchar: A tool to infer characteristics of Internet paths"*, ftp://ftp.ee.lbl.gov/pathchar

[6] Ji, Z., Yang, Y., Zhou, J., Takai, M., Bagrodia, R., "Exploiting Medium Access Diversity in Rate Adaptive Wireless LANs," MobiCom 2004.

[7] Kamerman, A. and Monteban, L., "WaveLAN II: A high-performance wireless LAN for the unlicensed band," Bell Labs Technical Journal, pp. 118-133, Summer 1997.

[8] Kapoor, R., Chen, L.-J., Lao, L., Gerla, M., Sanadidi, M. Y., "CapProbe: A Simple and Accurate Capacity Estimation Technique,"SIGCOMM 2004.

[9] Lacage, M., Hossein, M., and Turletti, T. IEEE 802.11 Rate Adaptation: A Practical Approach. In Proceedings of ACM MSWiM 2004.

[10] Lai, K., Baker, M., *"Measuring Bandwidth,"* IEEE INFOCOM '99

[11] Li, J., Blake, C., Couto, D., Lee, H. I., and Morris, R., "Capacity of Ad Hoc Wireless Networks," ACM MobiCom 2001.

[12] Mills, D. L. "Network Time Protocol Specification, Implementation and Analysis," RFC 1305, March 1992.

[13] Moon, S. B., Skelly, P., and Towsley, D., "Estimating and Removal of Clock Skew from Network Delay Measurements," IEEE Infocom 1999.

[14] Paxson, V., "On Calibrating Measurements of Packet Transit Times," ACM SIGMETRICS 1998.

[15] Qiao, D., Choi, S. Jain, A., and, S, K. G., "MiSer: An Optimal Low-Energy Transmission Strategy for IEEE 802.11a/h", MobiCom 2003.

[16] Sadeghi, S., Kanodia, V., Sabharwal, A., and Knightly, E., "Opportunistic Media Access for Multirate Ad Hoc Networks," MobiCom 2002.

[17] Zhang, L., Liu, Z., and Xia, C. H., "Clock Synchronization Algorithms for Network Measurements," IEEE Infocom 2002.

[18] OAR, *ww.ece.rice.edu/networks/software/OAR/OAR.html*

[19] NS-2 Simulator, www.mash.cs.berkeley.edu/ns/