Bandwidth Measurement in Wireless Mesh Networks

Zhenyu Yang
Department of Computer Science
University of Illinois at Urbana-Champaign
zyang2@uiuc.edu

Chandrakanth Chereddi
Department of Electrical Engineering
University of Illinois at Urbana-Champaign
cchered2@uiuc.edu

Haiyun Luo
Department of Computer Science
University of Illinois at Urbana-Champaign
haiyun@cs.uiuc.edu

ABSTRACT
In this project, we investigate the issue of measuring end-to-end bandwidth in wireless mesh networks. Bandwidth measurement is an active research topic in the networking area as the information of link capacity and available bandwidth is essential for the network management, traffic control, and QoS provisioning. Although quite a few research progresses have been made in the wired networks, the bandwidth measurement in the wireless mesh networks is yet to be explored. In the latter domain, we need to face totally new challenges including cross-traffic interference and multi-hop effect. To design and evaluate measuring tools, we have set up a testbed of wireless mesh network. Our goal is to analyze those factors via real testing environment and to propose relevant techniques on the basis of what we have observed from the experiments.

Keywords
wireless mesh network, bandwidth measurement

1. INTRODUCTION
The full (partial) mesh networks refer to a certain network topology where each node is connected to all (some of) other nodes. Combined with "wireless" and "ad hoc" techniques, the wireless mesh networks gain advantages of low cost, reliability, scalability, and adaptability, making them an appealing choice for the construction of community network to extend the Internet services. Although the benefits of reliability and adaptability are inherent in the mesh topology, those potentials are not realized spontaneously without a crucial step of accurately measuring the quality of wireless links. We have seen lots of research efforts and tools proposed for measuring bandwidth and other quality parameters of wired networks. However, the measurement issue in the wireless networks has so far received relatively less attention.

To classify some basic concepts, the network bandwidth can be measured on the basis of per *hop* or *end-to-end path*. For each hop/path, we can evaluate its *capacity* or *available bandwidth*. The capacity is the upper limit of the transmission rate while available bandwidth the spare room for accommodating more traffic. Thus, we have four different combinations of modeling along the two dimensions. In this project, we are interested in the end-to-end bandwidth capacity of wireless path. As a minor point, the measurements can be done in two modes, *active mode* and *passive mode*. In the passive mode, the bandwidth is estimated by capturing packets without injecting extra traffic to the network. It may be an important concern in the wireless network due to the shortage of bandwidth resources.

There are two major groups of measuring path capacity in the wired networks, namely the *variable packet size probing* (VPS) and *packet pair/packet train dispersion probing* (PPTD). VPS approaches the issue by exploiting the relationship of packet size and *round-trip time* (RTT). It sends packets of various size and uses linear interpolation to estimate the bandwidth. PPTD basically sends multiple probing packets back-to-back and compute the dispersion time between them. It is assumed that by sending packets at a suitable rate two packets will be queued adjacent to each other at the source of the slowest link and the dispersion will be maintained towards the end of the path. Therefore, PPTD actually measures the *bottleneck bandwidth* which is usually taken as a valid representative of the bandwidth capacity of the whole path in wired networks.

To start with, we are setting up a testbed of wireless mesh network to investigate those techniques that are borrowed from the wired networks. From some preliminary results, we have the following observations.

1.1 Metrics
Most of the tools only measure the individual hop or bottleneck bandwidth in a multi-hop path. Although such measurement is quite indicative of the bandwidth quality in the wired networks, we show by some simple examples that it is not as useful as in the wireless networks because of the interaction of hops sharing one transmission channel. In Figure 1, we have a wireless path of four hops. When the first hop is active in transmission, the second and third hop cannot transmit simultaneously. However, whether the fourth hop can transmit or not depends on the real situation of interference. Hence, even we have the knowledge of bandwidth capacity of each hop, the overall capacity of the path

is still unclear. Another example is shown in Figure 2 to

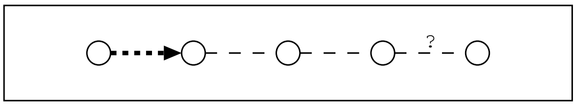

Figure 1: The Effect of Multiple Hops

demonstrate the effect of bottleneck link in both wired and wireless networks. For case (a), we suppose a 5-hop wired path where the fourth hop is the bottleneck. The figure shows three consecutive packets passing through the path. If we model the network using the concept of pipelining and assume perfect scheduling of packets along each hop to allow maximum overlapping of the transmission, it is obvious that the path capacity is determined by the bottleneck bandwidth instead of the topology. For case (b) and (c), we use wireless links and in both cases the bottleneck remains as the fourth hop. However, we make the bottlenect bandwidth of (c) smaller than that of (b). We adopt a simple interference model by specifying that the first and fourth hops can transmit simultaneously and so do the second and fifth hops. For example, when the first packet propagates to the fourth hop, the second packet can start transmitting through the first hop since there is no interference between the two hops and we assume perfect scheduling. We can see that, due to the interference, both wireless paths cannot be fully pipelined and although (c) has a smaller bottlenect bandwidth it actually has better overall bandwidth than that of (b). The simple example shows that the path capacity is more complicated in the wireless networks. The bottleneck bandwidth provides some hint but it is not as meaningful as in the wired networks.

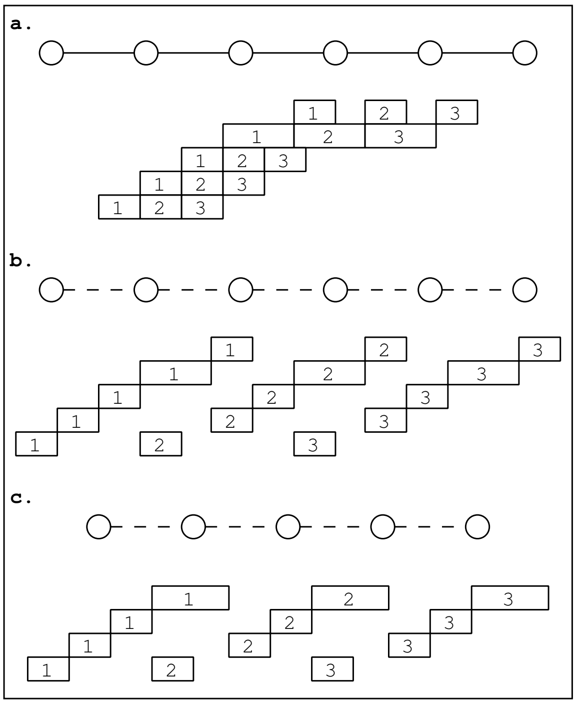

Figure 2: The Effect of Bottleneck Hop

1.2 Applicability

We find that tools based on packet pair/train usually do not perform in the wireless networks. They either provide inconsistent results or no results at all. The assumption those techniques based upon is that two packets can be queued at the source node of the bottleneck hop and after that hop there is no more queueing for the two packets. Otherwise, the dispersion time of the two packets generated by the bottleneck will be distorted when arriving at the destination. In wired network, it is feasible to collect good samples even under high load of traffic. However, such assumption is hardly possible in the wireless network. The queueing delay may not be the real issue but it is replaced by delays caused by the interference and contention which are hard to remove if we have to send two packets back-to-back.

1.3 Timeliness

Some of the tools take substantial amount time to converge. This may not be a serious issue in the wired network or in the community wireless networks where the topology tends to be more stable. However, a fast program is still preferrable for volative wireless networks or multi-path bandwidth probing.

1.4 Intrusiveness

We see that tools based on VPS push large number of packets into the network. Since the resource of bandwidth is relatively scarce in the wireless networks, the tools relying on sending huge number of probing packets will render themselves less attractive. We need to design tools with more reticent behaviors.

We deem that not only the techniques but also, more importantly, the target of metrics should be adapted. The accuracy of measurement remains a major criteria but the promptiveness and quietness emerge as valid concerns. With all those requiements, designing measuring tools for the wireless networks become more challenging.

The rest of the paper is organized as follows. In Section 2, we present some related work in measuring bandwidth of wired network, including some available techniques. Section 3 presents the results and analyses of those techniques applied in the wireless mesh network. In section 4, we perform extensive experiment to study the influence of hops and distances on the wireless channels. We propose a measureing technique in Section 5 and evaluate its performance in Section 6. We conclude the paper in Section 7 and discuss future work in Section 6.

2. RELATED WORK

We first examine the available measurement techniques in the wired network. Paper [1] provides a good survey of bandwidth measurement in several combinations. As mentioned ealier, the basic techniques of measuring bandwidth capacity are VPS (*variable packet size probing*, [7]) and PPTD (*packet pair/packet train dispersion probing*, [3]). In [2], an interesting arrangement of load packets and measurement packets is used in a single packet train to find the bottleneck along an end-to-end path. The measurement has a more realistic sense as the results reflect the influence of real traffic. For end-to-end available bandwidth measurement, *Self-loading periodic streams* (SLoPS, [10]) tries to match the sending rate with the available bandwidth and *Pathchirp* ([13]) uses the self-induced congestion principle. By far, per-link available bandwidth measurement has not been found in the literature.

There are several publicly available tools based on those techniques. For example, *Pchar* ([8]) and *Clink* ([12]) are based on VPS while *Nettimer* ([9]), *Pathrate* ([11]), and *CapProbe* ([3]) use packet pairs/trains to measure the end-to-end capacity.

3. EXPERIMENT

We set up the testbed of ad-hoc wireless network based on IEEE 802.11b with 11Mb/s. The testbed consists of four laptops to form a 3-hop path. By far, we have tested *Pchar*, *Clink*, *Nettimer*, *Capprobe*, *Pathrate*, and *Pathchirp*. Their general performance under one hop and multi-hop environment is listed in Table 1, followed by more detailed results and discussions.

Table 1: The General Performance of Tools

Tool	Performance
Pchar	end-to-end measurement with per hop capacity
Clink	end-to-end measurement with per hop capacity
Nettimer	one hop measurement, not working under multiple hops
Capprobe	inconsistent results under one hop
Pathrate	not working under one hop
Pathchirp	end-to-end measurement with per hop available bandwidth

3.1 Pchar

We use *Pchar* to measure the end-to-end capacity of a 3-hop path. It uses probing packets of size ranging from 32 to 1500 bytes. The results are shown in Table 2.

Table 2: The Results of Pchar

	1st hop	2nd hop	3rd hop
bandwidth (Mb/s)	5.65	4.17	4.15

Pchar gives pretty decent per hop bandwidth estimation (see our later discussion) even though the first hop tends to have higher value of bandwidth than the next two hops. However, it takes considerable amount time and probing packets for the measurement.

3.2 Clink

We use *Clink* to measure the end-to-end capacity of a 2-hop path. It uses probing packets of size ranging from 28 to 1500 bytes. For each hop, it shows a low estimate, a high estimate, and a best estimate as shown in Table 3. Note that the best estimate does not alway lie beteen the low and high estimate.

Compared with *Pchar*, *Clink* gives far less accurate estimation. However, its cost in time and numer of packets is at the same level as *Pchar*.

Table 3: The Results of Clink

	low	high	best
1st hop (Mb/s)	3.26	3.27	3.24
2nd hop (Mb/s)	2.92	3.02	3.13

3.3 Nettimer

We only obtain the estimation made by *Nettimer* under one hop path with the value around 4.5Mb/s. The result is also consistent with what they present in [3]. However, after running it a couple of times we find that the results bear large deviation ranging from over 10Mb/s to below 1Mb/s. It shows that the program does not have effective control of the error.

None of the other measuring tools based on packet pair/train can provide acceptable results even under one hop condition. For *Capprobe*, it generates fluctuating results in each probing without any dominant values while *Pathrate* terminates in the middle due to some programming error. It confirmes the fact that as it is difficult for the basic assumption to hold in the wirelss network, those tools cannot work properly.

3.4 Pathchirp

Unlike all other tools, *Pathchirp* measures the available bandwidth of end-to-end path. We test it on a 3-hop path which generates a list of estimations with timestamps. It starts with a highest value of 9.7Mb/s and gradually drops to around 0.2Mb/s in 300 seconds. Such kind of regular decrement makes us suspect of its validity.

3.5 Discussion

Fundamentally, VPS is a *single probing packet strategy*, meaning that it sends one probing packet at a time, measures the RTT, and sends the next packet. If each packet (of size L) goes through n hops and each hop k has the propagation dely α_k and serialization delay $\frac{L}{C_k}$ (as shown in Figure 3), the total amount of delay up to hop i is

$$D_i(L) = \sum_{k=1}^{i}(\alpha_k + \frac{L}{C_k}) = \alpha + \beta_i L$$

We assume α_k is constant and $\beta_i = \sum_{k=1}^{i} \frac{1}{C_i}$. Therefore, if we take samples of different packet size L and use linear interpolation, the gradient will show β_i which can be used to compute the per hop capacity C_k. One factor that could possibly compromise VPS is the queueing delay, which is usually handled by taken enough number of samples and some statistical analysis. Theoretically, VPS should perform equally well in the wireless networks because each probing only involves with one packet. There is indeed an extra form of delay which is caused by the cross-traffic interference. However, this could be handled in the similar fashion as well. Intuitively we can think of this as a more *general* form of queueing delay. The results of *Pchar* verifies this point. However, as we have argued earlier the knowledge of per hop capacity is not that useful in the wireless network. On the other hand, techniques based on PPTD encounters a more serious problem that cannot be easily smoothed out. It is a *multi-probing packet strategy* which means that in each probing at least two packets are involved. PPTD relies

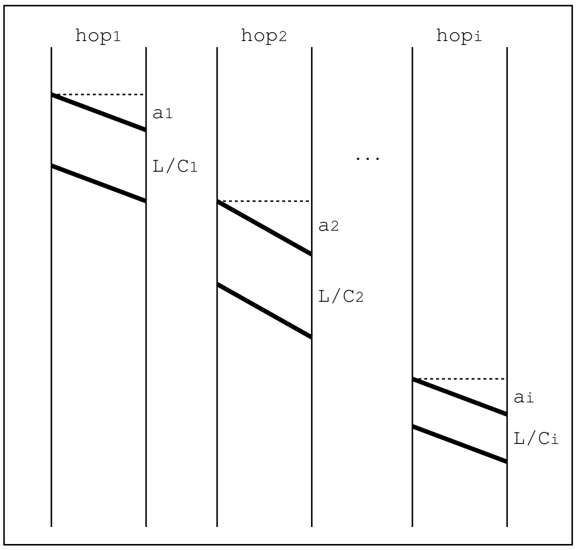

Figure 3: The Timeline of VPS

on sending multiple packets (of same size L) back-to-back and measuring the dispersion time between the last bit of the first packet and the last packet. Take packet pair for instance, after passing the first hop the dispersion time is $\frac{L}{C_1}$. Consider the second hop, if $C_2 \geq C_1$ then the dispersion after passing the second hop will maintain as $\frac{L}{C_1}$. Otherwise, it will become $\frac{L}{C_2}$. The second case is shown in Figure 4. Therefore, when the two packets arrive at the destination the dispersion will be $\frac{L}{C'}$ where C' is the bandwidth capacity. Queueing is the major factor that could destroy such technique. For example, when $packet_1$ is transmitted on hop_1 other packets can accumulate between $packet_1$ and $packet_2$ which adds to the dispertion time and causes underestimation. Likewise, in the wired networks this could be dealt with by statiscal analysis since they are also cross-traffic interferences.

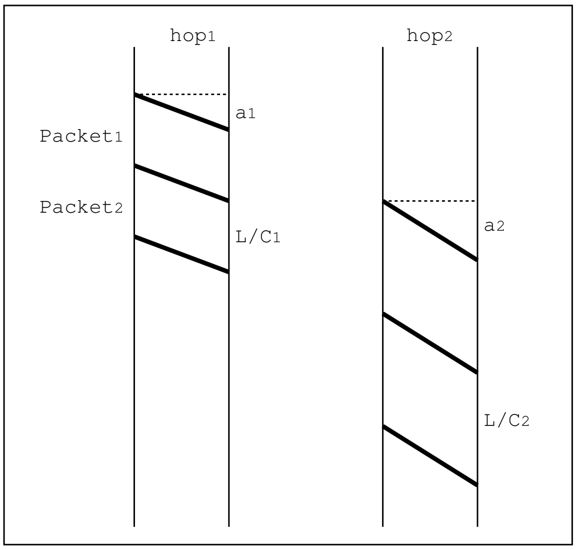

Figure 4: The Timeline of Packet Pair

4. EXPERIMENT OF CHANNEL PERFORMANCE

There is one important question we try to seek answers. What are the useful metrics to characterize the wireless channel? Some of our previous experiment results measuring the round-trip time of *ping* imply that the wireless channel has very dynamic nature. In Figure 5, we test *ping* of varying packet sizes with both one and two hops at the distance of one meter. As we can see, the mean round trip increases almost linearly in both cases, a certain kind of predictability. Figure 6 shows a similar experiment but with a larger distance of 20 meters. The observation is that as the distance and the number of hops are increased, the link becomes less predictable.

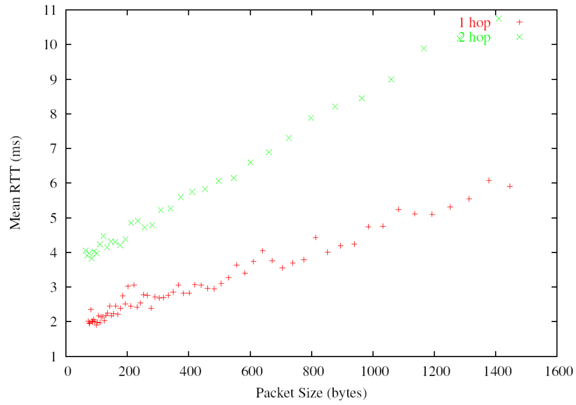

Figure 5: The Mean RTT Comparison of One Hop vs. Two Hops (1 meter)

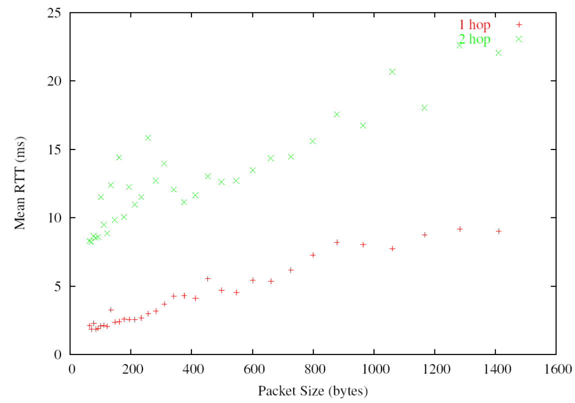

Figure 6: The Mean RTT Comparison of One Hop vs. Two Hops (20 meters)

In the next experiment, we use one computer to send UDP packets to another computer with fixed packet size (768 bytes) but varying the sending intervals. Figure 7 shows the loss ratio at the distance of one meter. As we increase the sending rate, the loss ratio is also increasing linearly and we can use the trend to estimate the available bandwidth pretty accurately. Figure 8 shows the similar experiment at the distance of 10 meters. However, the results become much less predictable. Figure 9 shows the results at the distance of 20 meters.

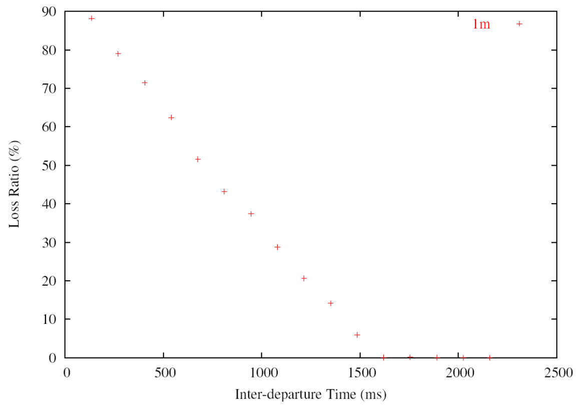

Figure 7: The Loss Ration with Varying Sending Rate (1 meters)

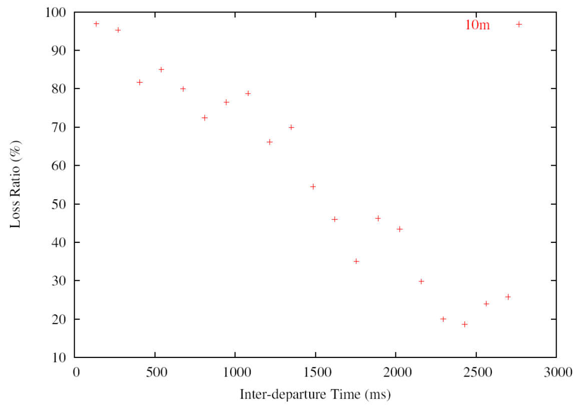

Figure 8: The Loss Ration with Varying Sending Rate (10 meters)

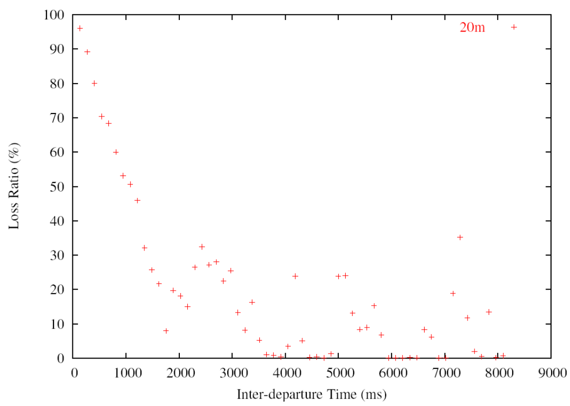

Figure 9: The Loss Ration with Varying Sending Rate (20 meters)

Our recent experiment exposes the other side of the wireless channel that may help us to find the answer. In each of the following experiments, we send a very large number (=100,000) of back-to-back packets (fixed packet size of 768 bytes) from one computer to another computer and collect the distribution of inter-arrival times. In the first experiment, we use two computers at the distance of one meter. Figure 10 shows the distribution of inter-arrival time. Although the inter-arrival time ranges from $10\mu s$ to $18250\mu s$, most of them are concentrated in the range of (1830, 1850), which counts for 77%. Figure 11 takes a more closer look at the concentrated range.

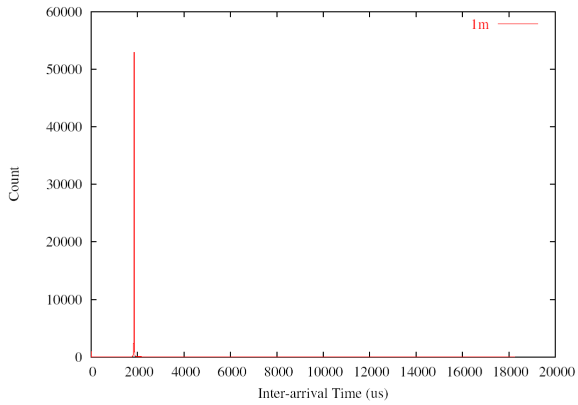

Figure 10: The Inter-arrival Distribution of Large Packet Train (1 meter)

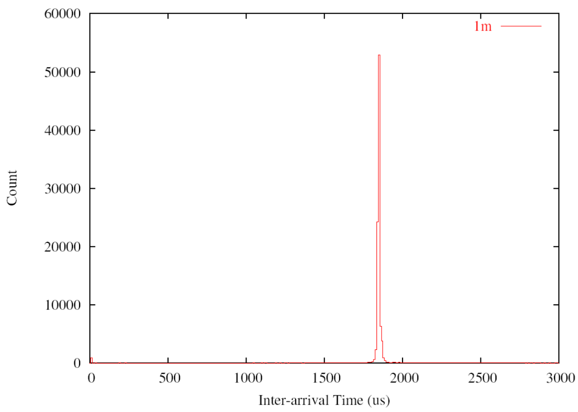

Figure 11: The Closer Look at Inter-arrival Distribution of Large Packet Train (1 meter)

Figure 12 shows the distribution of inter-arrival times of two computers at the distance of 20 meters. Expectedly, the distribution has a much wider range (10 - $74560\mu s$). However, the predominance of certain arrival time is still there with the concentrated range of (1830, 1850) counts for 65%. Figure 13 takes a more closer look around the concentrated range. It shows that there is another small peak around 2500.

Figure 14 shows the distribution of inter-arrival times of two computers at the distance of 30 meters. The range becomes even wider but the predominance is still obvious. However, we observe more small peaks by a closer look (Figure 15).

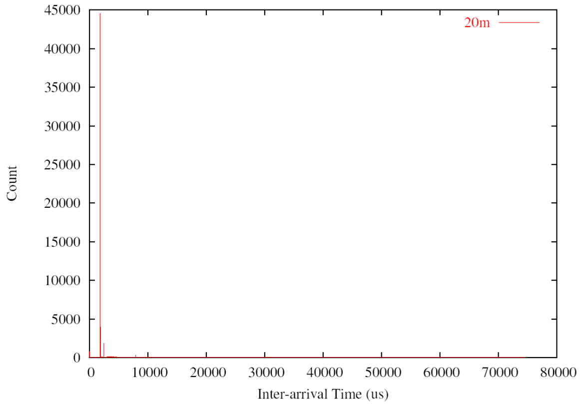

Figure 12: The Inter-arrival Distribution of Large Packet Train (20 meters)

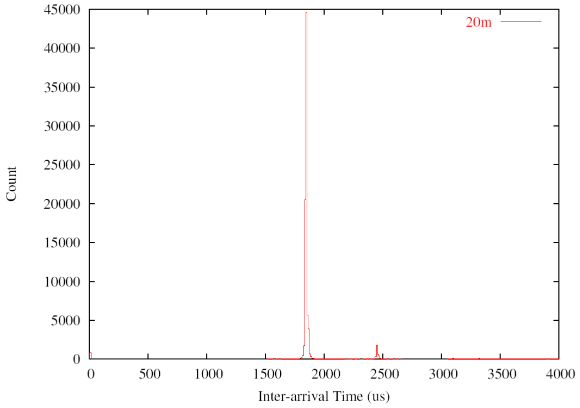

Figure 13: The Closer Look at Inter-arrival Distribution of Large Packet Train (20 meters)

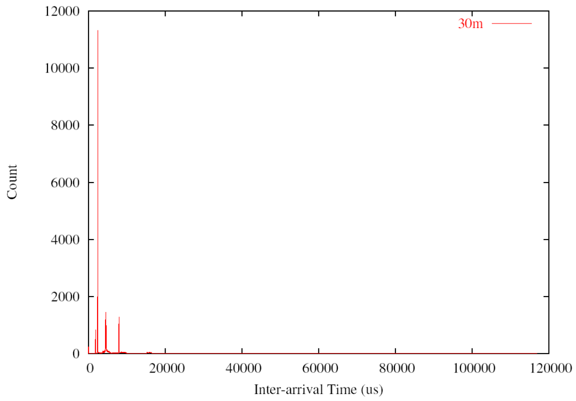

Figure 14: The Inter-arrival Distribution of Large Packet Train (30 meters)

The next experiment uses a multihop of three computers. Each hop has a distance of 10 meters. Figure 16 gives the overall distribution and Figure 17 magnifies the interested area. In the second plot, we see the predominance followed by some larger hills in the range of (2000, 3500). When two packets sent back-to-back through a 2-hop path, it has

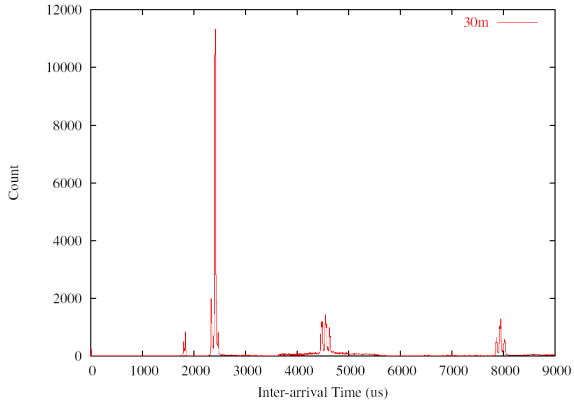

Figure 15: The Closer Look at Inter-arrival Distribution of Large Packet Train (30 meters)

two possibilities. For one case, they can be queued at the intermediate node such that the receiver only sees the inter-arrival time as the delay of one hop. For another case, the first packet can travel through the two hops before the second packet starts transmitting. Then the receiver sees the inter-arrival time as the delay of two hops. As mentioned earlier, the variation of two hops is much bigger than one hop. Hence, we see a predominance followed by a hill.

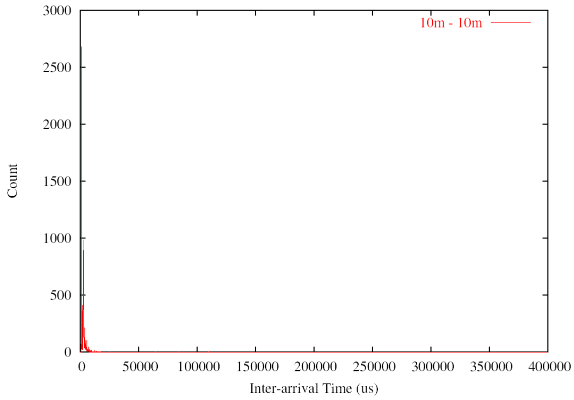

Figure 16: The Inter-arrival Distribution of Large Packet Train (10 meters - 10 meters)

Those experimental results show that the characterization of wireless channel is very complicated. The factors of distance and number of hops introduce dynamics and unpredictability. Our earlier results show the trends that we hope could lead to some single parameter for describing the channel state. Later on, we feel that it may not be a realistic goal. We are now seeking some statistical tools for modeling.

5. PROPOSED APPROACH

The idea of the approach is to combine the technique of *PPTD* with VPS. Packet pair technique gives a measure of goodput, which is the total useful application level bytes transferred as given in Equation 1.

$$g = \frac{s}{t} \tag{1}$$

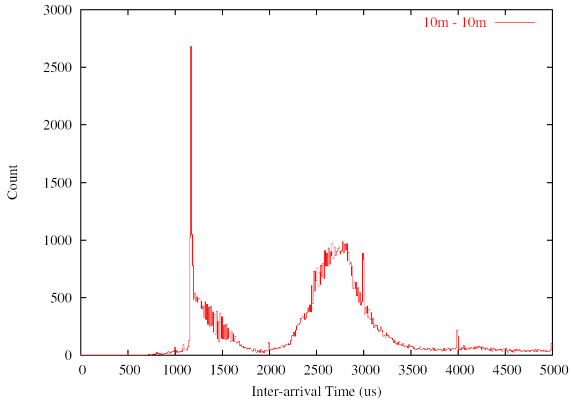

Figure 17: The Closer Look at Inter-arrival Distribution of Large Packet Train (10 meters - 10 meters)

where s is the packet size in bits, t the transmission time in seconds, and g the goodput in bits per second.

For the wireline netowrks, the *throughput* (T) is nearly equal to the *goodput* (g), i.e. $\frac{g}{t} > 0.95$. Therefore, g can be safely used as a measure of throughput. However, for the wireless networks, the application level throughput is never close to the channel level throughput. Even at the *maximum size data unit* ($MSDU$), the goodput for 802.11b networks at 11 Mbps approximates 6.1 Mbps. This means the utilization is only 60%. Even lower packet sizes, such as 100 bytes, will have very low utilization. Approximately, the overhead per packet in bytes is around 800 bytes.

Hence, we study the task of calculating what the overhead is. Given a multihop network, if a given node (any node) in the network either uses RTS/CTS or CSMA but does not have threshold based usage of RTS/CTS, then the throughput is constant. If an RTS/CTS fires at a node based on packet size, the overhead will be multi-modal (or bi-modal) based on the specific configuration.

We consider the overhead to be constant as denoted by x in the following equations and calculations. If x is overhead in bits, the bandwidth (i.e. throughput) at the channel level is given in Equation 2.

$$T = \frac{s + x}{t} \tag{2}$$

where s is the packet size, t the dispersion time of packet pair, T the throughput in mega bits per second.

Since x is constant over various packet sizes for a given network setup, it holds that.

$$T = \frac{s_1 + x}{t_1} \qquad T = \frac{s_2 + x}{t_2} \tag{3}$$

By some simple calculations, it can be shown that.

$$T = \frac{s_1 - s_2}{t_1 - t_2} \tag{4}$$

However, from our experiments we observed that using values of t_1 and t_2 from different packet train sizes makes the calculations very unstable. Therefore, we use values of g and then from g we can calculate the overhead and also throughput as in Equation 5.

$$g_1 = \frac{s_1}{t_1} \tag{5}$$

We collect many values of g_1, we then filter this data with a band pass near the median which allows all values of $g_1 \in [median(g_1) - range, median(g_1) + range]$. We also calculate the average value of g_1. The same is repeated for different packet sizes (g_1, g_2, g_3 ...). In our experiment, we uses packet sizes of 200, 500, 800, 1100, 1400 bytes.

Now,

$$\frac{s_1 + x}{t_1} = T = \frac{s_1 - s_2}{t_1 - t_2} \tag{6}$$

Replace t_1 with $\frac{s_1}{g_1}$ and $t - 2$ with $\frac{s_2}{g_2}$, we have,

$$\frac{s_1 + x}{\frac{s_1}{g_1}} = \frac{s_1 - s_2}{\frac{s_1}{g_1} - \frac{s_2}{g_2}} \tag{7}$$

which gives,

$$x = \left(\frac{s_1 - s_2}{\frac{s_1}{g_1} - \frac{s_2}{g_2}}\right) \times \frac{s_1}{g_1} - s_1 \tag{8}$$

Using combinations of packet sizes from s_1, s_2, s_3, s_4, s_5 as (s_1, s_2), (s_1, s_2), (s_1, s_3), (s_1, s_4), (s_1, s_5) etc, we get many values of x (overhead) in bits. The values obtained for x, are again filtered over a range, which was chosen at $(median(x) - \frac{median(x)}{4}, median(x) + \frac{median(x)}{4})$. The remaining values of x are averaged to get a final measure of x, the overhead in bits.

Next, we can calcuate throughput T as.

$$T_1 = \frac{s_1 + x}{s_1} \times g_1 \tag{9}$$

Similarly,

$$T_2 = \frac{s_2 + x}{s_2} \times g_2 \qquad T_3 = \frac{s_3 + x}{s_3} \times g_3 \quad \ldots \tag{10}$$

For a final measurement of T, we use average of T_1, T_2, T_3 ... T_n, or.

$$T = \frac{\Sigma T_i}{n} \tag{11}$$

Finally, given a value of T and overhead x, under our assumption we can calculate the goodput at packet size s as in Equation 12.

$$g = \frac{s}{s + x} \times T \tag{12}$$

6. TESTBED

To obtain the experimental results of the bandwidth measurement, we set up a testbed of multihop wireless networks in the Coordinated Science Laboratory.

6.1 Single Cell Multihop Network

In the first set of tests, we arrange the nodes in one cell such that each node in the network can reach any other node within one hop distance. However, we set up the routing table to force multihop routing. The topology of the given network and the routing path is given in Figure 18. The histogram of distribution of the bandwidth measure-

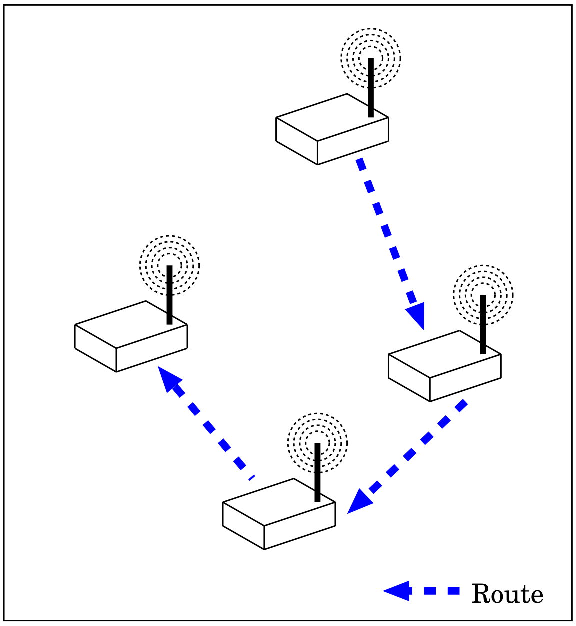

Figure 18: The Topology of A Single Cell Multihop Network

ment is given in Figure 19. The average bandwidth is 3.781

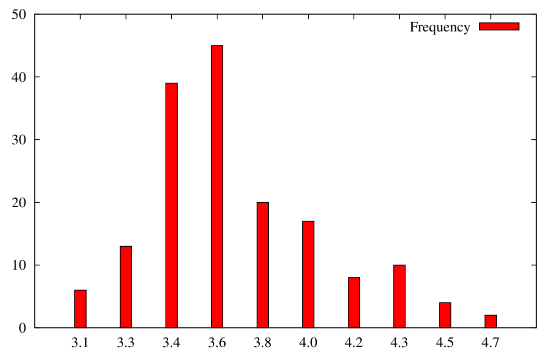

Figure 19: The Bandwidth Measurement of Three Hops in A Single Cell Network

Mbps with standard deviation of 0.345. The value is close to theorectical expection (i.e. $\frac{11}{3}Mbps$).

6.2 Multi-Cell Multihop Network

In the second set of tests, we arrange the nodes with hops of longer distance to form multiple cells. The topology of the given network and the routing path are given in Figure 20. Note that hop_{BC} is much longer than hop_{AB} and hop_{CD}.

The bandwidth measurement of multi-cell is presented in Figure 21. The average bandwidth is 4.249 Mbps with stan-

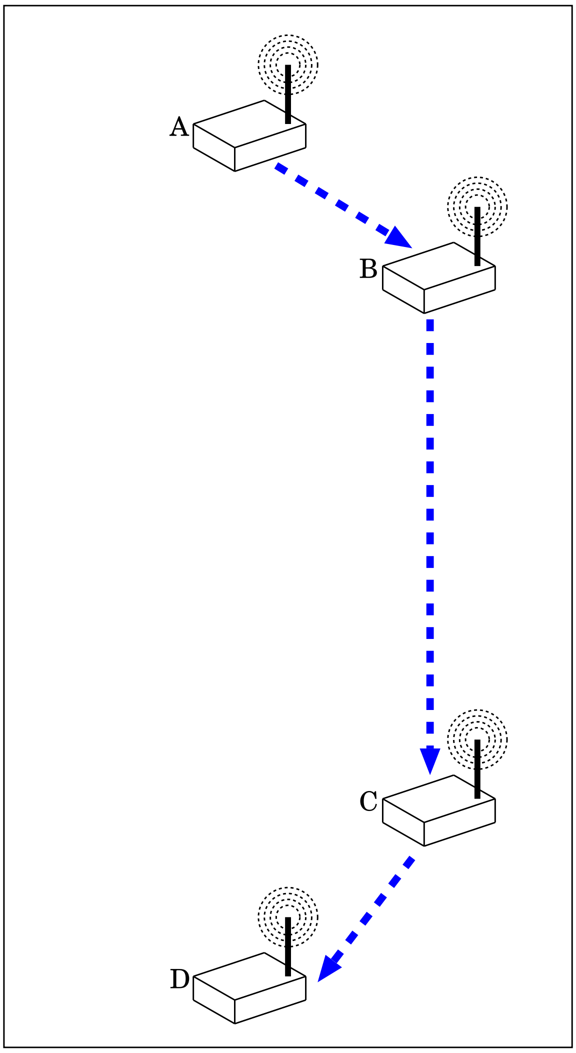

Figure 20: The Topology of A Multi-Cell Multihop Network

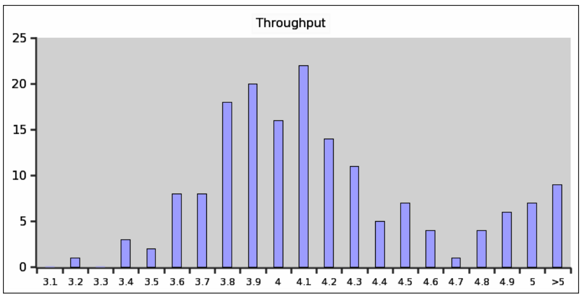

Figure 21: The Bandwidth Measurement of Three Hops in A Multi-Cell Network

dard deviation of 0.394. We can see as the hop distance increased the bandwidth is actually increased due to less interference.

We then study the single-hop performance under multi-hop environment as current research work attempts to use single-hop measurement to infer the multi-hop performance. We want to have some insights as how valid such approaches would be. We first present the bandwidth of one of the shorter hops (hop_{AB}) in Figure 22 followed by the bandwidth of the longest hop (hop_{BC}) in Figure 23. The average

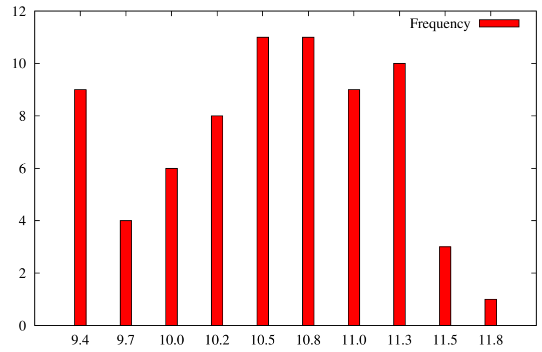

Figure 22: The Bandwidth Measurement of Hop AB

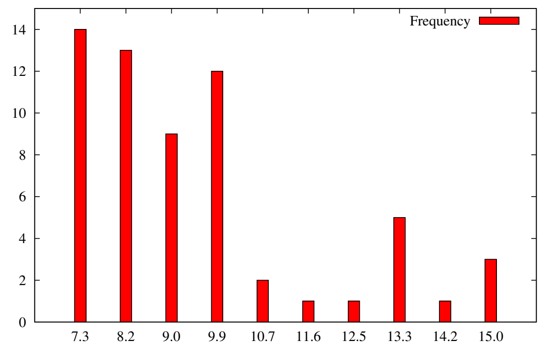

Figure 23: The Bandwidth Measurement of Hop BC

bandwidth of hop_{AB} is 10.647 Mbps with standard deviation of 0.637. The average bandwidth of hop_{BC} is 9.949 Mbps with standard deviation of 2.199. The overall trend is clear: as the distance is increased the bandwidth is reduced and becomes less predictable. For one hop communication, shorter distance is preferred. However, for multi-hop communication, it is better to separate nodes with larger distance. The trade-off here would be an interesting research topic. The results also indicate that it is unclear whether we can possibly infer the multi-hop performance from single-hop measurement. At least it is not straightforward by some simple calculations.

6.3 Packet Size Impact

We want to investigate the impact of packet size on goodput. Figure 24 shows the goodput of packet size of 500 bytes. The average goodput of 500 bytes packet size is 1.459 Mbps with standard deviation of 0.291.

Figure 25 shows the goodput of packet size of 800 bytes. The average goodput of 800 bytes packet size is 1.879 Mbps

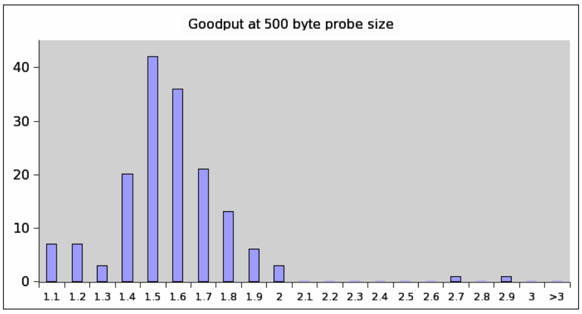

Figure 24: The Goodput of Packet Size 500 bytes

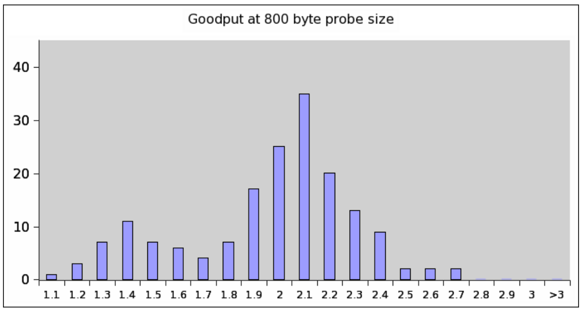

Figure 25: The Goodput of Packet Size 800 bytes

with standard deviation of 0.386.

Figure 26 shows the goodput of packet size of 1100 bytes. The average goodput of 1100 bytes packet size is 2.231 Mbps

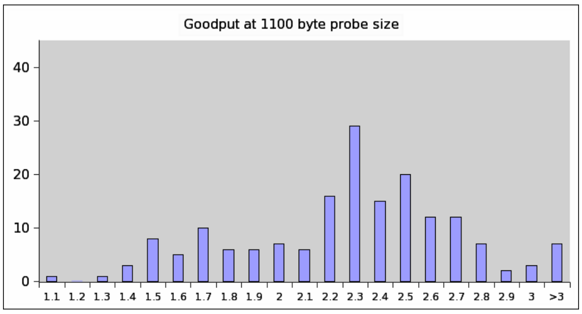

Figure 26: The Goodput of Packet Size 1100 bytes

with standard deviation of 0.501.

Figure 27 shows the goodput of packet size of 1400 bytes. The average goodput of 1400 bytes packet size is 2.443 Mbps with standard deviation of 0.454.

As the packet size is increased, the goodput is also increased. However, with the increase in packet size the cost of error such as packet loss is increased as well. That's why the distribution seems more scattered.

7. SUMMARY

Here is the brief summary of what we have done in the project. We start with analyzing the problem of bandwidth measurement in wired networks. We try several measuring

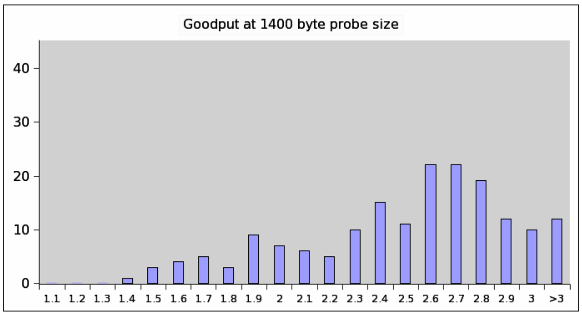

Figure 27: The Goodput of Packet Size 1400 bytes

tools based on the techniques of *variable packet size probing* (VPS) and *packet pair/packet train dispersion probing* (PPTD). None of these tools we have tested provides satisfatory results. We further study them and give the reason of their poor measuring performance. Later on, we carry out extensive experiment to observe the influence of different combination of hops and hop distances on the network performance. Our goal is to establish some modeling of the network behavior. The results give us lots of interesting insights which could serve as a starting point for further investigation.

We propose a measuring technique combining PPTD and VPS. The drawback of PPTD is that it omits overhead measurement while VPS assume symmetric links. Our method includes the consideration of overhead measurement and applies to asymmetric links. We then evaluate the technique by running it on a real testbed of multihop wireless networks.

8. FUTURE

Throughout the whole project, we keep trying to find some answer to a problem which seems easy at the first glance. However, the deeper we dig into it, the more difficult it turns out. Although we have achieved some results, there are lots of questions left unanswered.

- For the proposed measurement technique, we want to understand the impact and relation of train size and packet size on the measurement. Our fundamental goal is to get accurate measurements with as less traffic injected as possible. But what is the optimal combination?
- The effect of cross traffics is not analyzed. In our experiment, we have no knowledge of the background noise. What is the amount of cross traffics? What are their patterns? How do they affect the measurement? A good measurement should reflect the variation of the realtime traffic. Without the knowledge of background traffic, it impossible to make such evaluation. However, it is technically hard unless we can fully control the environment. Otherwise, we may fall into a logical loop.
- Although it is somewhat convincing that the single-hop measurement is not closely correlated to the multihop performance. It may be still worthwhile to find some weak relations between the two to derive some coarse measuring models. The reason is that single-hop measurement is relatively straightforward and less expensive.
- Unexpectedly, the establishment of wireless testing environment becomes a formidable task. We have very limited resources for setting up large-scaled wireless mesh networks to perform more realistic tests. Unlike research on wired networks where machines can be packed in a small space and experiments can be synchronized by a central controller, research on wireless networks is much more painful by just booting a correct testing case. We conceive the construction of a testbed that could facilitate the research, for example, to deploy wireless devices in several selected offices in the building or to access some commercial roofnet that may be available.
- The ultimate question is still left unanswered. That is, what are the metrics? If the measurement tells that one multihop path is better than other paths, how can a client verify the claim? If he runs some applications, he may get different conclusions. One important observation from the project is that the traffic pattern in wireless networks has a great effect on the actual bandwidth achievable. Maybe we can take a look at passive measurement as well since fundamentally all we know and all we can tell is how a network is currently being used. However, what we observe by injecting a probing sequence is actually the aggregated effect of yet another traffic. Therefore, if the user's application traffic is very different from our probing patterns, he will definitely get different conclusions.

Our work, as we always feel, is rather a beginning than an end. We gain more knowledge and experience of wireless networks before taking on the project. This will definitely have a positive impact on our own individual research work as bandwidth measurement is so fundamental in the networking area. We appreciate this opportunity to work together on something meaningful and interesting.

9. REFERENCES

[1] R. S. Prasad, M. Murray, C. Dovrolis, and K. Claffy, *Bandwidth Estimation: Metrics, Measurement Techniques, and Tools*, IEEE Network, Vol. 17, No. 6, pages 27-35, November 2003.

[2] N. Hu, L. Li, Z. M. Mao, P. Steenkiste, and J. Wang, *Locating Internet Bottlenecks: Algorithms, Measurements, and Implications*, Proceedings of ACM SIGCOMM 2004, August 2004.

[3] R. Kapoor, L.-J. Chen, L. Lao, M. Gerla, and M. Y. Sanadidi, *CapProbe: A Simple and Accurate Capacity Estimation Technique*, ACM SIGMETRICS Performance Evaluation Review, Vol. 32, No. 1, June 2004.

[4] R. German and A. Heindl, *Performance evaluation of IEEE 802.11 wireless LANs with stochastic Petri nets*, Proceedings of 8th International Workshop on Petri Net and Performance Models (PNPM'99), pages 44-53, October 1999.

[5] A. A. Bhat, *Stochastic Petri Net Models of Service Availability in a PBNM System for Mobile Ad Hoc Networks*, Master Thesis in Computer Engineering, Virginia Polytechnic Institute and State University, July 2004.

[6] C. Rust, F. Stappert, and T. Lukovszki, *A Petri Net Model for the Network Layer of a Mobile Ad Hoc Network Architecture*, 7th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2003), July 2003.

[7] V. Jacobson, *Pathchar: A Tool to Infer Characteristics of Internet Path*, ftp://ftp.ee.lbl.gov/pathchar/, April 1997.

[8] B. A. Mah, *Pchar*, http://www.ca.sandia.gov/ bmah/Software/pchar/, 2000.

[9] K. Lai and M. Baker, *Nettimer: A Tool for Measuring Bottleneck Link Bandwidth*, Proceedings of the USENIX Symposium on Internet Technologies and Systems, March 2001.

[10] M. Jain and C. Dovrolis, *End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput*, Proceedings of ACM SIGCOMM, August 2002.

[11] C. Dovrolis, P. Ramanathan, and D. Moore, *What do Packet Dispersion Techniques Measure?*, Proceedings of IEEE Infocom, April 2001.

[12] A. B. Downey, *Clink: A Tool for Estimating Internet Link Characteristics*, http://allendowney.com/research/clink/.

[13] V. Ribeiro,R. Riedi,R. Baraniuk,J. Navratil, and L. Cottrell, *pathChirp: Efficient Available Bandwidth Estimation for Network Paths*, Passive and Active Measurement Workshop 2003.