TSProbe: A Link Capacity Estimation Tool for Time-Slotted Wireless Networks

Ling-Jyh Chen[1], Chih-Wei Sung[2], Hao-Hsiang Hung[1], Tony Sun[3], and Cheng-Fu Chou[2]

[1]Institute of Information Science, Academia Sinica
[2]Department of Computer Science and Information Engineering, National Taiwan University
[3]PacketMotion, Inc.

***Abstract*— In this paper, we propose TSProbe, a new capacity estimation technique for time-slotted wireless systems. Although inspired by Adhoc Probe, the design of TSProbe is tailored to time-slotted connections. Unlike legacy approaches that rely on probes of identical size, TSProbe takes advantage of the intrinsic relationships between various link layer properties, and deploys an adaptive and iterative probing scheme that is both efficient and accurate. We analyze the theoretical basis of TSProbe, and evaluate its performance in a variety of simulation scenarios. The results show that the technique is consistently accurate and effective in all cases.**

I. Introduction

Estimating the link capacity of a path connection is a fundamental research problem in computer networking because knowledge of the link capacity is crucial to the efficient design, management, and usage of a network. A number of capacity estimation techniques have been proposed in recent years [4] [8] [9] [13] [14] [16] [21], and among them, CapProbe [14] and Pathrate [13] are the two most widely used techniques. Several works have demonstrated that these two techniques are fast and accurate in both wired and last-mile wireless scenarios [14] [15] [18] [19]. Even so, some studies have shown that, in emerging multihop wireless networks, CapProbe and Pathrate fail to estimate the link capacity correctly due to the dynamics of the wireless environment (e.g., node mobility, route changes, wireless interference, and other environment sensitive factors) [11]. To resolve this problem, two CapProbe variants, namely AdHoc Probe [11] and SenProbe [22], have been proposed to estimate end-to-end path capacity in IEEE 802.11-DCF-based wireless ad hoc networks (using CSMA/CA with an RTS/CTS multiple access scheme) and CSMA-based (or CSMA/CA-based) wireless sensor networks.

Nevertheless, despite the popularity of CSMA-based medium access control schemes, many wireless network technologies (e.g., IEEE 802.15.1, IEEE 802.16, and IEEE 802.15.4) favor time-slotted multiple access schemes. A detailed understanding of the issues involved in estimating channel capacity in time-slotted wireless networks is therefore important. Not only does capacity estimation in time-slotted wireless networks need to withstand wireless interference and other environmental factors, it is also essential that the estimation scheme can fully accommodate the concept of time slotted communication. More specifically, since nodes are assigned "busy" or "idle" slots in time-slotted multiple access schemes, over-estimation of the link capacity may occur if the channel capacity is calculated according to the burst data transfer rate during a "busy" time slot, whereas under-estimation may occur if the channel capacity is calculated without fully utilizing the "busy" time slot. Therefore, appropriate consideration of the "busy" and "idle" time slots must be the first step towards better utilization of the link capacity.

In this paper, we study the issues related to link capacity estimation and link utilization in time-slotted wireless networks. Using our Bluetooth testbed, we first evaluate the performance of several popular capacity estimation tools for time-slotted wireless systems, and discuss the shortcomings of existing approaches. We then analyze the relationships between link capacity estimation, channel utilization, and the following system parameters: the application's packet size, the IP layer's header size, the data link layer's header size, the basedband packet size, and the maximum data rate of the wireless channel. Based on our findings, we propose a new capacity estimation scheme, called TSProbe, for time-slotted wireless systems. The technique was inspired by Adhoc Probe. However, instead of relying on identically sized payloads, TSProbe employs an iterative probing scheme that utilizes payloads that vary in size. Specifically, TSProbe takes advantage of the intrinsic relationships between various link layer properties, and uses different-sized payloads to intelligently probe for the baseband packet size. We evaluate TSProbe in a variety of simulation scenarios, the results of which demonstrate the efficacy of the scheme. The implications of this study are not limited to Bluetooth networks; they are applicable to all the other emerging time-slotted wireless network technologies.

The remainder of the paper is organized as follows. Section II summarizes related works on capacity estimation and briefly reviews the Bluetooth technology. In Section III, we discuss the experiment results of applying several popular link capacity estimation tools on Bluetooth links, and use a simple analytical model to explain how channel utilization affects path capacity estimates. In Section IV, we describe TSProbe, the proposed link capacity estimation tool for time-

This work is sponsored by the National Science Council under grant numbers NSC-94-2218-E-001-002.

slotted networks. Using NS-2 simulations, we evaluate the performance of TSProbe in multihop Bluetooth networks in Section V. Then, in Section VI, we present our conclusions.

II. Background and Related Work

A. Link Capacity Estimation

Until recently, research on capacity estimation focused on delay variations among probe packets, as in Pathchar [8], or on the dispersion of probe packets, as in Nettimer [16] and Pathrate [13]. However, Pathchar-like tools (such as pchar [9] and clink [4]) are limited in terms of their accuracy and speed, as reported in [14] [17]. Moreover, they evaluate the capacity of a link based on the estimates of previous links along the path, so that estimation errors are propagated as the length of the path increases [21].

Dispersion-based techniques are prone to other problems. In particular, Dovrolis' analysis in [13] shows that the dispersion distribution can be multi-modal due to cross traffic, and that the strongest mode of the distribution may correspond to (1) the capacity of the path; (2) a "compressed" dispersion, resulting in over-estimation of the link capacity; or (3) the Average Dispersion Rate (ADR), which is always lower than the link capacity. Another dispersion-based tool, SProbe [21], exploits SYN and RST packets of the TCP protocol to estimate the downstream link capacity, and employs two heuristics to filter out samples affected by cross traffic. However, SProbe is not effective when the network utilization is high [19].

In contrast to the above approaches, CapProbe [14] uses both dispersion and end-to-end delay measurements to filter out packet pairs distorted by cross traffic, and it has been shown to be both fast and accurate in a variety of scenarios. Recent capacity estimation studies have extended the target network scenarios to more diverse environments. For instance, ABLP [20] and AsymProbe [12] have been proposed for estimating the capacity of increasingly popular asymmetric links, while [11] and [22] extended CapProbe to estimate the *effective path capacity* in ad hoc wireless networks and wireless sensor networks.

B. Bluetooth

Bluetooth is a short-range, low cost, and low power consumption radio technology that operates on the unlicensed 2.4GHz ISM (Industrial-Scientific-Medical) frequency band. It uses the Frequency Hopping Spread Spectrum (FHSS) and implements stop and wait ARQ (Automatic Repeat reQuest), CRC (Cyclic Redundancy Check), and FEC (Forward Error Correction) techniques to ensure high reliability on the wireless links and alleviate the effects of interference from other radio wave emitting technologies (i.e., 802.11b [5], cordless phones, and microwave ovens).

Bluetooth units can be connected to other Bluetooth units to form a piconet that can support up to eight active units. One of the units acts as a master and the others act as slaves. All data/control packet transmissions are coordinated by the master. Slave units can only transmit in the slave-to-master slot after being addressed in the preceding master-to-slave slot. Each slot lasts for 625 microseconds.

TABLE I

Basic Data Types in Bluetooth ACL Mode

Mode	FEC	Packet Size (bytes)	Packet Length (slots)	Symmetric Throughput (Kbps)	Asymmetric Throughput (Kbps)	
DM1	Yes	17	1	108.8	108.8	108.8
DM3	Yes	121	3	258.1	387.2	54.4
DM5	Yes	224	5	286.7	477.8	36.3
DH1	No	27	1	172.8	172.8	172.8
DH3	No	183	3	390.4	585.6	86.4
DH5	No	339	5	433.9	723.2	57.6

For real-time data, such as voice content, Synchronous Connection Oriented (SCO) links are used, while Asynchronous Connectionless Links (ACL) are used to transmit other data. ACL packets differ in length, which affects the data transmission rate, and they may or may not use FEC coding. Table I lists the basic ACL packet types and their respective properties. Note that in the asymmetric connection mode, the Bluetooth link occupies 1/3/5 time slots (for DM1/DM3/DM5 or DH1/DH3/DH5 modes) in one direction, but only one time slot in the opposite direction. In this paper, we focus on the asymmetric connection mode used by the Bluetooth ACL link because it is employed by majority of Bluetooth networks.

III. Estimating Time-Slotted Link Capacity

A. Comparison of popular capacity estimation tools

In the first set of experiments, we use our Bluetooth testbed to evaluate the accuracy and speed of several popular link capacity measurement tools. Although, for simplicity, we focus on one-hop Bluetooth links, the results of this study can be easily extended to multihop scenarios. We believe the experiment results from multihop scenarios would be similar to those of the one-hop scenario presented in the next subsection (assuming the inter-piconets scheduling mechanisms are known a priori).

In the experiments, we employed two link capacity estimation techniques (AdHoc Probe [11] and Pathrate [13]) and a *Bulk Transfer Capacity* (BTC) measurement tool (Iperf [6]). Using two Linux-configured laptops, the experiments were conducted on top of matching DLink DBT-122 Bluetooth adapters. AdHoc Probe's settings were configured to probe the Bluetooth connection with 1500-byte packets (including the UDP and IP header overheads) at a rate of 5 probes per second. For the Iperf and Pathrate experiments, we used the default packet sizes (i.e., 1498-byte packets for Iperf in UDP mode, 1500-byte packets for Iperf in TCP mode, and adaptive packet sizes for Pathrate) and the default probing rates. Each configuration was tested 10 times. The average results from the 10 runs are shown in Tables II and III.

The test results show that the BTC techniques are efficient in that, generally, they require less than 20 seconds to estimate the link capacity; however, the results are not accurate in all test cases. Meanwhile, the link capacity estimates of AdHoc

TABLE II

THE AVERAGE LINK CAPACITY ESTIMATES OF A ONE-HOP BLUETOOTH LINK. (UNIT: KBPS)

Tool	Bluetooth Packet Type					
	DH5	DH3	DH1	DM5	DM3	DM1
Theoretical	723.2	585.6	172.8	477.8	387.2	108.8
AdHoc Probe	643	541	138	459	370	84
Pathrate	608	537	137	432	360	84
Iperf - UDP	541	469	129	415	328	82
Iperf - TCP	539	462	126	397	321	80

TABLE III

THE AVERAGE TIME TAKEN TO OBTAIN THE LINK CAPACITY ESTIMATE FOR A ONE-HOP BLUETOOTH LINK. (UNIT: MM'SS)

Tool	Bluetooth Packet Type					
	DH5	DH3	DH1	DM5	DM3	DM1
AdHoc Probe	0'40	0'40	0'40	0'40	0'40	0'40
Pathrate	18'33	18'41	18'38	18'41	18'43	0'34
Iperf - UDP	0'12	0'12	0'17	0'13	0'13	0'22
Iperf - TCP	0'11	0'12	0'16	0'12	0'12	0'20

Probe and Pathrate are generally close to the theoretical maximum data throughput of a one-hop Bluetooth link, but the accuracy varies from 78% to 95%, depending on the type of Bluetooth baseband packet employed.

Additionally, we observe that Pathrate requires an excessive amount of time to estimate the link capacity in almost all test cases, except the DM1 mode. This is because the Pathrate algorithm first probes the network with different-sized packets (the first phase) in search of a unimodal distribution. If a unimodal can not be found, Pathrate probes the network again using the second phase of the algorithm, which is a very intrusive and time consuming process. In our test scenarios, we found that Pathrate actually enters the second probing phase in all test cases, except DM1 packets. Since we do not consider cross traffic in our experiments, the Pathrate results indicate that capacity estimates are affected by the size of the probing packets. We discuss this issue in the following subsection.

B. Analysis

We now analyze the effect of the size of the probing packets on capacity estimates and channel utilization. For simplicity, cross traffic and packet loss are not considered. In addition, we assume that the baseband packet type of the Bluetooth link is the same for all experiments, such that the maximum data throughput and the baseband packet size are fixed. We denote the maximum data throughput of the baseband packet type as C, the size of each baseband packet as L, the size of the Bluetooth stack header as H, and the probing packet size as P(including UDP and IP headers). In most cases, $P \gg L$; therefore, each probing packet must be fragmented into several small baseband packets before transmission. The number of baseband packets required to transmit one probing packet is $\lceil \frac{P+H}{L} \rceil \times L$. The capacity estimate can then be obtained by:

$$E = \frac{P}{\lceil \frac{P+H}{L} \rceil \times L} \times C \qquad (1)$$

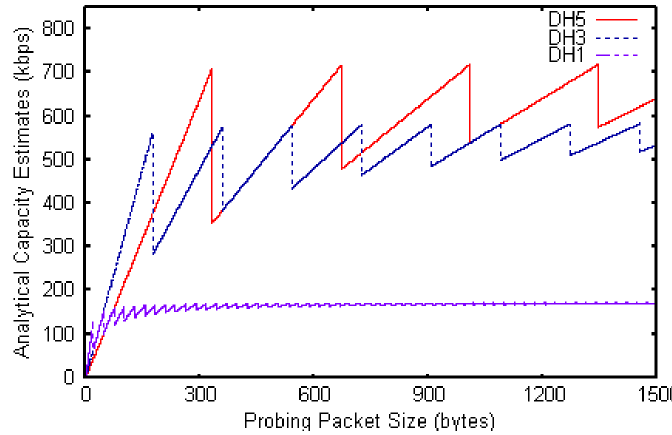

Fig. 1. Analytical AdHoc Probe capacity estimates for one-hop Bluetooth links.

where $\frac{P}{\lceil \frac{P+H}{L} \rceil \times L}$ represents the overall "utilization" of the channel. Clearly, channel utilization depends on the size of the baseband packet and on how well a probe packet fragments into baseband packets.

The value of H, which is based on the Bluetooth spec [1], can be observed experimentally by using the BlueZ '*hcidump*' utility program [2] to monitor baseband packets transmitted on a Bluetooth connection. In our experiments, we found that the value of H was 7 bytes, comprised of a 4-byte L2CAP header and a 3-byte BNEP header [3]. We use this H value, as well as the L and C values listed in Table I to plot the analytical capacity estimates of Bluetooth DH1/3/5 modes against various P values, as shown in Fig. 1.

The results show that the capacity estimate oscillates as the size of the probing packets changes. More specifically, the "oscillation cycle" corresponds to the baseband packet size, L, and the "oscillation range" decreases as the probing packet size increases. In other words, in time-slotted systems like Bluetooth, **the *effective* link capacity is dependent on the packet size**. This means the maximum link capacity can only be obtained if the probing packet size (including the header of each layer) is a multiple of the baseband packet size. If a less than ideal packet size is used (one that is not a perfect multiple of the baseband packet size), the channel capacity cannot be fully utilized, which in turn degrades the maximum link capacity estimate.

IV. PROPOSED APPROACH: TSPROBE

To address the shortcomings of existing methods, we propose a new link capacity estimation tool, called TSProbe, for time-slotted networks. Although TSProbe was inspired by AdHoc Probe, it is designed specifically for estimating the link capacity of a time-slotted network. Unlike legacy approaches, which use identical probe packets, TSProbe considers the link layer's properties and employs an iterative and adaptive probing scheme that utilizes different-sized packets.

More precisely, the rationale for TSProbe is based on the following: if R denotes the ratio of E (capacity estimate) to

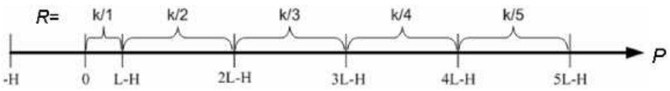

Fig. 2. The TSProbe algorithm. R is a step function, k/n, where $k = C/L$; n is the minimal number of baseband packets required to transmit one IP layer packet of size P; and $nL - H$ denotes the sizes of the packets that yield the peak value in Fig. 1.

P (payload size), and k denotes the ratio of C (connection capacity) to L (baseband packet size) (i.e., k is a constant with a value equal to C/L), then R can be derived by the following equation:

$$R = \frac{E}{P} = \frac{C}{\lceil \frac{P+H}{L} \rceil \times L} = \frac{k}{\lceil \frac{P+H}{L} \rceil} \quad (2)$$

Since $\lceil \frac{P+H}{L} \rceil$ is also a constant that denotes the minimal number of baseband packets required to transmit one IP layer packet (of size P), R can be treated as a step function k/i ($i = 1, 2, 3, ...$) with a constant L, as shown in Fig. 2.

¿From Fig. 2, it is clear that the interval value L (i.e., the basedband packet size) can be derived as long as the sizes of two IP layer packets, $p[0]$ and $p[1]$, satisfy Eq. 3, 4, and 5. These three equations are carefully designed conditions that ensure $L = abs(p[0] - p[1]))$, where $p[0]$ and $p[1]$ yield two adjacent local maximums. The algorithm used to find L in TSProbe is detailed in Alg. 1. Adhoc Probe is used to facilitate the algorithm's probing. To fulfill the conditions stated in Eq. 3, 4, and 5, probing packets with different payload sizes are carefully selected in an iterative manner based on feedback from the previous packet's payload size. As a result, the TSProbe estimation scheme is not only fast and accurate, it also has the simplicity and reliability of Adhoc Probe.

$$\lceil \frac{p[0] + H}{L} \rceil = \lceil \frac{(p[0] + 1) + H}{L} \rceil - 1 \quad (3)$$

$$\lceil \frac{(p[1] + 1) + H}{L} \rceil = \lceil \frac{p[0] + H}{L} \rceil \quad (4)$$

$$\lceil \frac{p[1] + H}{L} \rceil = \lceil \frac{p[0] + H}{L} \rceil + 1 \quad (5)$$

Once L has been obtained, the link layer header size, H, can be calculated as follows:

$$H = L - (p[0] \ \ mod \ \ L) \quad (6)$$

If $E[i]$ is the link capacity estimate of AdHoc Probe using $p[i]$ probing packet size, the maximum achievable data throughput (i.e., the theoretical link capacity) $C[i]$ can be derived by Eq. 7 (using Eq. 1). TSProbe then calculates the link capacity estimation results by averaging $C[0]$ and $C[1]$ (as shown in Eq. 8).

$$C[i] = \frac{E[i] \times L \times \lceil \frac{p[i]+H}{L} \rceil}{p[i]} \quad (7)$$

$$C = \frac{C[0] + C[1]}{2} \quad (8)$$

Algorithm 1 The algorithm for finding the baseband packet size, L, in TSProbe.

```
Function find_L (p_start, init_range)
for i = 0 to 1 do
  d ← init_range
  p_end ← p_start − d
  ratio1 ← AdhocProbe(p_start)/p_start
  while d > 1 or d < −1 do
    ratio2 ← AdhocProbe(p_end)/p_end
    if ratio1 == ratio2 then
      p_end ← p_end − d
      if p_end < 0 then
        d ← 0 − init_range
        if i == 1 then
          p_start ← p[0] + 1
        end if
        p_end ← p_start − d
      else
        p_start ← p_start − d
      end if
    else
      d ← int(d/2)
      p_end ← p_end + d
    end if
  end while
  if d > 0 then
    p[i] ← p_end
  else
    p[i] ← p_start
  end if
  p_start ← p[i]
end for
return abs(p[0] − p[1])
```

Note that TSProbe not only correctly estimates the link capacity in time-slotted networks in an end-to-end manner, it also derives the link layer header size and the baseband packet size. Moreover, once the values of L and H are known, it is easy to determine the optimal IP layer packet size P' (including the IP header size, which is usually 20 bytes)[1], where $P' = nL - H$ and n is the largest possible integer that can ensure P' is never larger than the MTU.

V. Evaluation

The simulations were carried out on an NS-2 network simulator [7] with the UCBT [10] extension. The initial simulation parameters were set as $p\_start = 1500$ bytes, $init\_range = 128$ bytes, and MTU = 1500 bytes. In addition, two network topologies were employed in the evaluation, namely (a) a piconet topology and (b) a scatternet topology (as shown in Fig. 3). In the piconet topology, the second node acts as the master node, and the others are slaves;

[1]By optimal packet size, we mean the packet size that yields the maximum data throughput in the given time-slotted network.

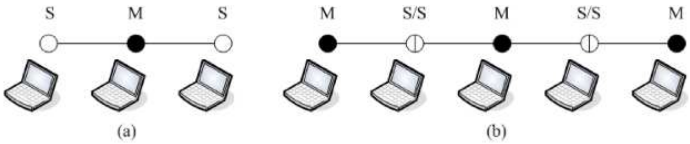

Fig. 3. The Bluetooth network topologies used in the simulations: (a) the piconet topology, and (b) the scatternet topology.

TABLE IV

SIMULATION RESULTS OF TSPROBE ON A ONE-HOP BLUETOOTH LINK WITH DIFFERENT TRANSMISSION MODES.

Mode	$p[0]$	$E[0]$	$p[1]$	$E[1]$	L	H	$C[0]$	$C[1]$	C
DH5	1349	719.0	1010	718.1	339	7	722.8	723.1	722.92
DH3	1457	583.8	1274	582.2	183	7	588.6	585.4	585.51
DH1	1478	171.5	1451	172.7	27	7	172.3	173.5	172.89
DM5	1337	476.0	1113	475.3	224	7	478.5	478.2	478.35
DM3	1445	385.8	1324	385.3	121	7	387.7	387.3	387.50
DM1	1489	108.7	1472	108.9	17	7	109.2	109.5	109.33

TABLE V

THE AVERAGE LINK CAPACITY ESTIMATES FOR THE BLUETOOTH NETWORK SCENARIOS USING TSPROBE.

Mode	Piconet (Kbps)		Scatternet (Kbps)		
	1 hop	2 hop	2 hop	3 hop	4 hop
DH5	722.92	361.46	351.13	361.46	361.46
DH3	585.51	292.75	266.08	292.75	292.75
DH1	172.89	86.44	83.79	86.44	86.44
DM5	478.35	239.18	244.76	239.18	239.18
DM3	387.50	193.75	180.56	193.75	193.75
DM1	109.33	54.66	51.70	54.66	54.66

whereas in the scatternet topology, the 1st, 3rd and 5th nodes are master nodes, and the others are slave/slave gateways that connect two adjacent piconets. We used the piconet topology for simulations with distances up to 2 hops, and the scatternet topology for distances between 2 and 4 hops. In each simulation, six Bluetooth baseband packet types were employed, and the results were gathered by taking the average of 10 runs.

Table IV shows the simulation results for TSProbe on a one-hop Bluetooth link with different baseband packet types. The results demonstrate that TSProbe can estimate the baseband packet size and the link layer packet header size in an end-to-end manner (i.e., without a priori knowledge of the link layer's properties). Note that the results are consistent with the Bluetooth specification presented in Table I and subsection III-B. The results also show that the capacity estimates (C) of TSProbe are much closer to the theoretical results of Bluetooth (shown in Table I) than the capacity estimates of other tools (shown in Table II). This is because the size of the probing packets in TSProbe is adapted to the estimated link layer's properties, whereas legacy tools only use one probe size.

We also performed simulations on multihop Bluetooth connections using the scenarios in Figures 3-a and 3-b scenarios. The results, detailed in Table V, show that the capacity estimates are very accurate for the 1-hop scenario. However, in multihop scenarios, the accuracy of the capacity estimates is approximately half that of the one-hop capacity estimates. This is because, in multihop Bluetooth networks, only one master-slave connection can be active at a time in each piconet. As a result, there is a risk that all "odd" links will operate simultaneously, and all "even" links will operate in parallel. In this case, since each link can only occupy half of the wireless channel at most, the theoretical link capacity is half that of the 1-hop link capacity, which validates our simulation results.

VI. CONCLUSION

In this paper, we consider issues related to link capacity estimation and utilization in time-slotted wireless networks. Using a Bluetooth testbed, we first evaluate the performance of several popular capacity estimation tools in time-slotted wireless systems. Based on our findings, we then analyze the relationship between link capacity estimates, channel utilization, and several system parameters (e.g., the application's packet size, the IP layer's header size and the data link layer's header size, the baseband packet size, and the wireless channel's capacity). Moreover, we propose a new technique called TSProbe, which estimates the link layer's properties and the link capacity of a time-slotted system in an end-to-end manner. The simulation results demonstrate the efficacy of the proposed technique. Work on evaluating TSProbe in other types of time-slotted wireless networks is ongoing. We will report on the results in the near future.

REFERENCES

[1] Bluetooth v1.2. http://www.bluetooth.com.
[2] Bluez. http://bluez.sourceforge.net.
[3] Bnep ver. 1.0. http://www.bluetooth.com.
[4] Clink. http://allendowney.com/research/clink/.
[5] Ieee 802.11, the working group setting the standards for wireless lans. http://grouper.ieee.org/groups/802/11/.
[6] Iperf. http://dast.nlanr.net/Projects/Iperf/.
[7] Ns-2. http://www.mash.cs.berkeley.edu/ns/.
[8] Pathchar. ftp://ftp.ee.lbl.gov/pathchar/.
[9] pchar. http://www.kitchenlab.org/www/bmah/Software/pchar/.
[10] Ucbt. http://www.ececs.uc.edu/c̃dmc/ucbt/ucbt.html.
[11] L.-J. Chen, T. Sun, G. Yang, M. Y. Sanadidi, and M. Gerla. Adhoc probe: Path capacity probing in ad hoc networks. In *WICON*, 2005.
[12] L.-J. Chen, T. Sun, G. Yang, M. Y. Sanadidi, and M. Gerla. End-to-end asymmetric link capacity estimation. In *IFIP Networking*, 2005.
[13] C. Dovrolis, P. Ramanathan, and D. Moore. What do packet dispersion techniques measure? In *IEEE Infocom*, 2001.
[14] R. Kapoor, L.-J. Chen, L. Lao, M. Gerla, and M. Y. Sanadidi. Capprobe: A simple and accurate capacity estimation technique. In *ACM SIGCOMM*, 2004.
[15] R. Kapoor, L.-J. Chen, M. Y. Sanadidi, and M. Gerla. Accuracy of capacity estimates using passive and active approaches. In *IEEE ISCC*, 2004.
[16] K. Lai and M. Baker. Measuring bandwidth. In *IEEE Infocom*, 1999.
[17] K. Lai and M. Baker. Measuring link bandwidths using a deterministic model of packet delay. In *ACM SIGCOMM*, 2000.
[18] K. Lakshminarayanan, V. N. Padmanabhan, and J. Padhye. Bandwidth estimation in broadband access networks. In *IMC*, 2004.
[19] S.-J. Lee, P. Sharma, S. Banerjee, S. Basu, and R. Fonseca. Measuring bandwidth between planetlab nodes. In *PAM*, 2005.
[20] Y. Lin, H. Wu, S. Cheng, W. Wang, and C. Wang. Measuring asymmetric link bandwidths using a multi-packet delay model. In *IEEE ICC*, 2003.
[21] S. Saroiu, P. K. Gummadi, and S. D. Gribble. Sprobe: A fast technique for measuring bandwidth in uncooperative environments. In *IEEE Infocom*, 2002.
[22] T. Sun, L.-J. Chen, G. Yang, M. Y. Sanadidi, and M. Gerla. Senprobe: Path capacity estimation in wireless sensor networks. In *SenMetrics*, 2005.