AdHoc Probe: End-to-end Capacity Probing in Wireless Ad Hoc Networks

Ling-Jyh Chen[1], Tony Sun[2], Guang Yang[2], M. Y. Sanadidi[2], and Mario Gerla[2]

[1]Institute of Information Science, Academia Sinica

[2]Department of Computer Science, University of California at Los Angeles

Abstract

Knowledge of end-to-end path capacity is useful for video/audio stream adaptation, network management and overlay design. Capacity estimation in wired and last-hop wireless networks has been extensively investigated, but a thorough and systematic study in ad hoc, multihop wireless networks is still lacking. Yet the rate of a wireless link can change dynamically (and rapidly) due to changes in interference, distance or energy optimization policy. Timely knowledge of path capacity is key to efficient routing, traffic management and application deployment. In this paper, we present AdHoc Probe, a packet-pair based technique, to estimate end-to-end path capacity in ad hoc wireless networks. We apply AdHoc Probe to path capacity estimation in auto rate wireless networks with variable displacement and interference; and, in remote wireless networks across the Internet. Using analysis, simulation and testbed experiments, we show AdHoc Probe can withstand mobility and is able to trace the rate adaptation of wireless networks timely and correctly. AdHoc Probe is simpler, faster and much less intrusive than current schemes.

Keywords: Ad hoc path capacity estimation; applications; analytical/simulation and experimental validation

I. Introduction

With the increasing deployment of wireless devices (e.g., laptops, PDAs, cellphones, etc), ad hoc networking is becoming an increasingly important class of infrastructure less technology for connecting a group of wireless devices. Ad hoc wireless protocols have been extensively investigated at all layers from physical to applications. However, a systematic development of ad hoc wireless network tools is still

Corresponding author: Ling-Jyh Chen (cclljj@iis.sinica.edu.tw)
Institute of Information Science, Academia Sinica.
Address: 128, Sec. 2, Academia Road, Taipei 11529, Taiwan
Tel: +886-2-2788-3799 ext. 1702; Fax: +886-2-2782-4814.

lacking. In particular, as a difference from the Internet, there are no efficient end-to-end tools to evaluate ad hoc network resources (e.g., path capacity, available bandwidth, etc.). Yet, the end-to-end knowledge of resources such as path capacity is important for network utilization and management. For instance, in a video conference application supported by an "overlay" that spans wired and wireless ad hoc users, the knowledge of path capacity to different destinations helps the sources and proxies adapt the audio/video streaming rates to match user capacities and provide better quality of services [3] [5]. A simple and accurate end-to-end path capacity estimation tool is needed. The estimation must be fast so that it can reflect the path capacity in a timely even when the actual capacity is varying (for example, because the user is moving from one media to another, or the environment interference is changing). The estimation must be independent of cross traffic (as in this case we are interested in evaluating the equivalent of the "bottleneck" capacity in the Internet, as opposed to "available" bandwidth). The estimation tool must be applicable to mixed wired and wireless paths, since several applications (especially the commercial applications) will include ad hoc wireless extensions as "opportunistic" extensions of the Internet. Finally, the estimation must be non-intrusive so that it will not disturb the ongoing applications traffic in the network.

The path capacity estimation problem has been extensively studied in the Internet. Various tools have been proposed to accurately estimate bottleneck link capacity in the wired and/or last-hop wireless networks [6] [9] [12] [14]. However, the complexity and convergence time required for these schemes are not well suited for ad hoc wireless networks. Moreover, the bidirectional set up of some of the above techniques proves to detrimental in ad hoc networks as discussed later.

Several techniques exist for adhoc path capacity estimation, which are supported by specific routing algorithms. For example, on demand routing algorithms (e.g., AODV [4]) as well as proactive algorithms (e.g., LANMAR and Fisheye Routing [24]) have been successfully extended to yield path capacity. These estimations however are dependent on specific routing schemes - they require feedback from the network layer. Our goal is to develop end-to-end estimation techniques independent of the network layer. This is critical in overlay applications, for example, where the source across the Internet does not know which specific routing scheme runs in the remote ad hoc network being probed.

Li et al directly addressed the end-to-end estimation of path capacity of a static multihop network based on 802.11b. Their approach was to send a brute force UDP packet stream and measure the maximum achievable data throughput [16]. This scheme is very realistic in that it reflects the currently available capacity if a UDP stream is injected. However, as discussed later it is not very practical due to impact on ongoing traffic. Moreover, the capacity measurement is affected by current cross traffic conditions. Other work, such as [15], evaluates capacity and available bandwidth of an end-to-end connection in last hop wireless scenarios, but not including wireless multihop scenarios.

End-to-end path capacity estimation in ad hoc wireless networks is a much more challenging problem than in wired nets. The estimation results need to be consistent with/without cross traffic, since the path capacity is a property that is invariant to the presence of cross traffic. At the same time, the estimate depends not only on the rate of the "narrow" link along the path (as in a wired net), but also on topology, path layout, interferences between nodes along the path and on several other environmental parameters. Moreover, the estimation technique must be applicable to both fixed rate and auto rate wireless networks (where rate can be adjusted to propagation characteristics and energy requirements). In other words, the estimate must account for rate adaptation and reflect it in a transparent way, with no feedback or side information from the network.

Motivated by the above goals, in this study we propose AdHoc Probe, a packet-pair based end-to-end technique that estimates path capacity in ad hoc wireless networks. We evaluated AdHoc Probe in static and mobile networks, with fixed and auto rate modems using simulation and testbed experiments. In all cases we show that AdHoc Probe can estimate path capacity timely and correctly.

The rest of the paper is organized as follows. In Section II, we summarize related work and recap the Internet CapProbe technique on which AdHoc Probe is based. An in-depth description of AdHoc Probe follows in Section III. In Section IV, we present an analysis of the path capacity in various adhoc wireless network configurations. Next, using NS-2 simulation, we validate AdHoc Probe in those configurations; namely, we address fixed rate wireless networks, including multihopping, relocation and mobility, in Section V; we extend the results to auto rate modems in Section VI. In Section VII, we present testbed experiment results that validate fixed and auto rate modem simulations. Finally, Section VIII concludes the paper.

II. Related Work and Background

A. Related Work

Link capacity estimation has been extensively studied in wired networks, e.g., [6] [9] [12] [14]. Previous research on capacity estimation relied on either delay variations among probe packets as illustrated in pathchar [9], or dispersion among probe packets as described in Nettimer [14] and Pathrate [6]. Conceptually, Dovrolis' analysis in [6] clearly revealed that the dispersion distributions can indeed be multi-modal without multi-channels, and that the strongest mode in the multimodal distribution of the dispersion may correspond to either (1) the capacity of the path, or (2) a "compressed" dispersion, resulting in capacity over-estimation, or (3) the Average Dispersion Rate (ADR), which is lower than the capacity.

In [12], Kapoor et al proposed a packet-pair based approach called CapProbe. CapProbe combines delay and dispersion measurements to accurately estimate bottleneck link capacity. [12] also compared the performance of CapProbe and the most representative technique, Pathrate, in terms of accuracy and speed in wired and last-hop wireless networks. From the comparison, both schemes are able to accurately estimate link capacities, but CapProbe outperforms Pathrate in estimation speed in the last-hop wireless scenario.

However, all of the above schemes have been and evaluated in wired and last-hop wireless scenarios. They have never been tested in ad hoc wireless networks. Capacity estimation in ad hoc wireless networks remains challenging due to the rapidly varying channel conditions, presence of node mobility, and multiple hops of wireless links.

In [16], Li et al examined the interaction of 802.11 MAC and ad hoc forwarding and the ability to infer path capacity for several simple configurations and traffic patterns. A brute force approach (i.e., using UDP flow to probe the maximum throughputs of the network) was used in simulation and experiments to validate the hypothetical limits of utilization in a chain network (1/4 of effective capacity). However, this approach was only able to obtain a substantially lower utilization of 1/7 of effective capacity.

In addition, as we mentioned earlier, Li's approach is intrusive and the result approximates path capacity only if the network is idle and static.

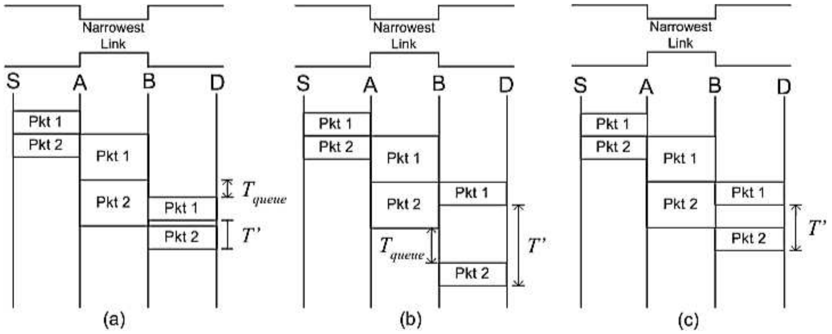

Fig. 1. Packet-pair technique: (a) Over-estimation caused by *compression* (b) Under-estimation caused by *expansion* (c) The ideal case.

B. CapProbe

CapProbe [12] is a recently proposed bottleneck capacity and path rate estimation technique shown to be both fast and accurate over a large range of scenarios. When two back-to-back packets are launched into a network, they are always separated at the bottleneck link by an interval related to bottleneck capacity. If such interval is preserved until destination unperturbed, it will allow us to compute the bottleneck link capacity (as shown in Fig. 1-c). The bottleneck capacity is equal to the UDP stream rate that can be sustained by the path - hence the name of “path rate” estimator often given to these tools. The interval between “interfered” packet-pair samples might be either expanded or compressed, where “expansion” leads to under-estimation and “compression” leads to over-estimation of the capacity (as shown in Fig. 1-a,b).

CapProbe combines the use of time interval measurements and end-to-end delay measurements to filter out packet pair samples interfered by cross traffic. By construction, an incorrect value of capacity estimate can occur only if cross traffic has interfered with the packet pair. In this case, queueing ensues and the delay of one or both packets will be larger than the minimum observed in absence of cross traffic. The sum of the delays of the packets in the packet pair is defined as the *delay sum*. A delay sum, which does not include any cross-traffic induced queuing delay, is referred to as the *minimum delay sum*. Any sample with delay sum larger than the observed minimum is thus discarded as it must have been interfered with. The capacity is easily derived from the equation:

$$C = \frac{P}{T} \tag{1}$$

where P is the sampling packet size, and T is the interval between packets with minimum delay sum.

CapProbe has been validated in wired and last-hop wireless networks with a variety of configurations. However, it has not been tested in ad hoc wireless networks yet. In wired networks, CapProbe is generally used in round trip mode, to evaluate the min capacity over the two directions. In ad hoc wireless networks, the round-trip mode is inadequate, since the first packet once relayed by the receiver may collide with the incoming second packet. Based on the CapProbe concepts, we thus design a one-way technique, called AdHoc Probe, to estimate the unidirectional path capacity (or, more properly, rate) in ad hoc wireless networks. We will present the algorithm and evaluations in the following sections.

III. AdHoc Probe

AdHoc Probe is based on CapProbe, a well-proved bottleneck link capacity estimation tool for wired and last-hop wireless networks [12]. However, AdHoc Probe differs from CapProbe in many significant ways. First, AdHoc Probe is a one-way (instead of round-trip) estimation technique. Secondly, AdHoc probe measures the maximum rate achievable on an "unloaded" path (i.e., no other users present) when intermittent environmental problems (e.g., short range mobility, random errors, etc) are factored out. It turns out that the achievable rate is generally considerably less than the min link capacity (while the two values are identical on a wired path). Thirdly, AdHoc Probe is designed to work under conditions not present in a typical Internet path, for instance when the wireless network is mobile, multihopped, interfered, and subject to rapid changes in link data rates.

In the following subsections, we present the AdHoc Probe algorithms in III-A and discuss the system time synchronization issue in III-B. In III-C, we study another potential system issue (i.e., the clock skew problem, which is critical in one-way network measurements) and present an efficient solution to it.

A. AdHoc Probe Algorithms

Similar to CapProbe, 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. Such metric can be used in end-to-end applications with QoS requirements, overlay designs, network traffic management, etc. 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

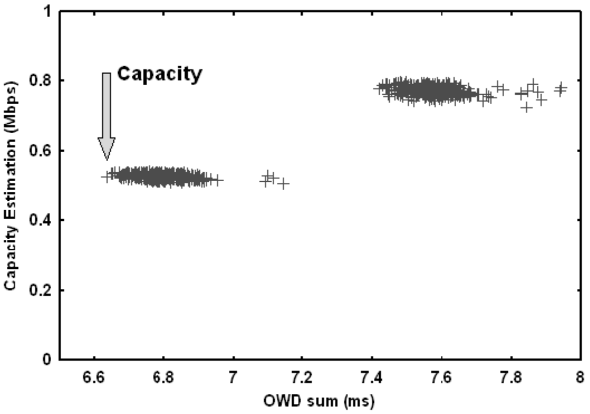

Fig. 2. AdHoc Probe capacity estimate using the sample with minimum OWD sum.

multihopping, 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 AdHoc Probe algorithm can be obtained by modifying the round-trip CapProbe into a one-way 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 time received (clocked at the receiver) and time sent (stamped 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 Fig. 2), and the corresponding capacity is given by Eq. 1.

Unlike CapProbe, AdHoc Probe does not implement the “convergence test” feature 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 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[1].

[1] When the network is highly dynamic (i.e., the nodes move fast, the wireless interference is severe, the route changes frequently, etc.), AdHoc Probe needs a large N in order to obtain an accurate estimate; however, AdHoc Probe may thus not be able to obtain the timely path capacity information because of the enlarged estimation latency. In such a case, it is recommended to run AdHoc Probe every few seconds with a reasonable size N (say, 200) and take the mean estimate (or the mode estimate that is the most frequent value among all estimates) as the estimation results.

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 of one estimation 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 tradeoff clearly depends on the application. In this paper, we set $N = 200$, $P = 1500$, and $R = 4$ sample pairs/sec for all simulations and testbed experiments.

B. System Time Synchronization Issue

The OWD measurement in AdHoc Probe is 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 [17]) 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:

$$S'_i = (T_{recv1,i} - T_{send,i}) + (T_{recv2,i} - T_{send,i}) \tag{2}$$

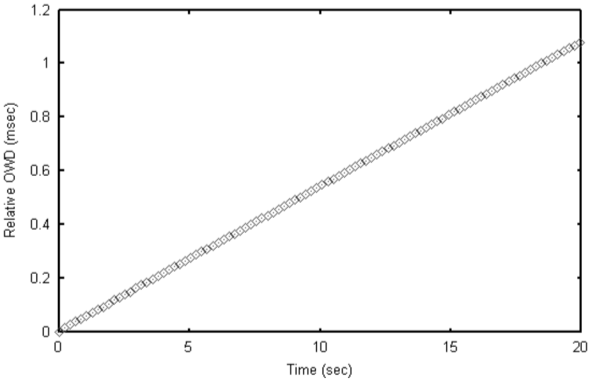

Fig. 3. Illustration of clock skew problem in OWD sum measurements.

$$
\begin{aligned}
S_i &= (T_{recv1,i} - T_{send,i} - \delta) + (T_{recv2,i} - T_{send,i} - \delta) \\
&= S'_i - 2\delta
\end{aligned}
\tag{3}
$$

Thus the difference between S_i and S'_i is a constant 2δ for all packet pairs. If $k = arg(min(S_i))$ for $i = 1...n$, then $S'_k = min_{i=1...n} S'_i$, 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.

C. Clock Skew Issue

In addition 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. Fig. 3 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) is skewed by almost 1 ms 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 Fig. 3). As a result, AdHoc Probe tends to select early (late) sample as the "good" sample when the OWD measurements are affected by an increasing (decreasing) skew.

Fortunately, the skewed clock drift problem has been efficiently solved in [25] by calibrating the skew.

We have implemented the “correction” in our code and have reported the details in the Appendix.

IV. What does AdHoc Probe actually measure?

A major difference between CapProbe and AdHoc Probe is the interpretation of the result. While in a wired network the path rate is equal to the bottleneck capacity, the path rate of a wireless multihop path is related to, but not necessarily identical, to the minimum of the link rates along the path. In the sequel we first review existing models that predict the effective rate and then show that AdHoc Probe indeed estimates such rate.

We recall that the effective end-to-end rate is defined as the maximum achievable data rate in the absence of any cross traffic connection. As mentioned earlier, this is smaller than the raw data rate at the physical layer. The difference is due to packet overhead and channel access coordination to handle multiple, pipelined packets on the path. We derive specific models below.

More specifically, in the 802.11b standard, since a RTS packet is 20 bytes, CTS and ACK packets are 14 bytes, and the MAC header of a data packet is 34 bytes, the effective capacity of a one-hop wireless link can be calculated by using the following equation[2]:

$$C = \frac{S}{S + 20 + 14 + 34} \times C_P \tag{4}$$

where S is size of the network layer packet (including IP header), and C_P is the link capacity of the physical layer. For instance, when the data packet size is 1,500 bytes and the data rate of the wireless link is 2Mbps, the effective capacity is at most $\frac{1500}{1500+20+14+34} \times 2 \approx 1.9$ Mbps.

However, due to the collision avoidance mechanism provided by RTS/CTS exchanges, the effective capacity of the wireless link decreases when there is more than one node within the same collision domain. This is clearly our case where the two packets are attempting transmissions within the same connection and path. It will be true also when a UDP stream will be maintained on the path. Suppose for instance that on the path in question there are $N-1$ active nodes within node A’s transmission range, all engaged in forwarding packets on the same path. The maximum effective rate for that path is C/N since only one of the N nodes can transmit a packet at a time. Naturally, it is unusual to have an ad

[2] In fact, the control frames (RTS, CTS, ACK) are sent using different transmission rate than the data frame, and the physical layer preamble and PLCP header require additional 192 us for transmission (which is equivalent to 22 bytes overhead in 1Mbps transmission rate). However, for simplicity, we assume all frames (RTS, CTS, ACK, DATA) are sent with the same transmission rate, and the physical layer overhead is negligible.

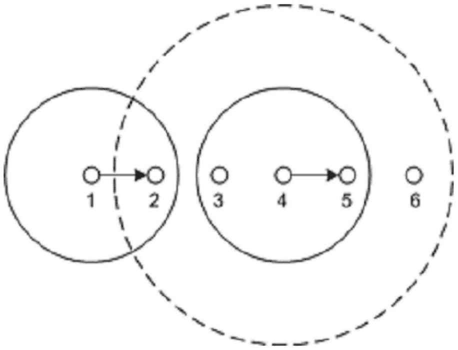

Fig. 4. The chain topology, where the solid-line circle denotes a node's effective transmission range and the dotted-line circle denotes a node's interference range.

hoc network path that hops several times within the same collision domain. However, this would clearly cause a reduction in effective rate. Such rate reduction must be captured by AdHoc Probe.

Much more common is the reduction in capacity incurred when the path is multihop. We consider a simple chain topology as shown in Fig. 4. For simplicity, we suppose the nodes are placed on a line with 200 meters to its immediate neighbor node, and the effective transmission range of each node is 250 meters. When the radio interfering range is the same as the transmission range, previous study by Li et al [16] has shown that the effective capacity of a chain topology becomes just 1/3 of the effective capacity of a single cell connection.

Moreover, as identified in [23], the radio interfering range is usually much larger than the transmission range. Therefore, the effective end-to-end capacity of a chain configuration will further decrease. For instance, in Fig. 4, if the interference range (marked by a dotted-line circle) is 550 meters, a packet transmission from node 4 will interfere with a packet transmission from node 1 to 2. In other words, simultaneous data transmission is not possible among nodes 1, 2, 3, and 4. It turns out that the effective end-to-end capacity of the chain topology in Fig. 4 will be at most 1/4 of the effective capacity of a single cell topology.

Another issue in a multihop path is that data rates may be different hop by hop due to different environment conditions. Thus, the link rate is no longer uniform along the path. In this case, the effective rate can still be computed with the above models, using the minimum rate link along the path.

From the above we can conclude that the effective rate in an ad hoc wireless path is more complex to model than in the wired network counterpart. One important feature of AdHoc Probe is that it does measure the correct path rate no matter how complex the underlying channel model (a physical system) is.

In the following sections, we will validate AdHoc Probe by comparing the path capacity estimation with the analytical results using simulation and testbed experiments. We will also study the path capacity of wireless networks in more diversified scenarios, e.g., where a link can change its rate according to the network conditions.

V. Simulation Results of Fixed Rate Wireless Networks

This section presents simulation results illustrating the used of AdHoc Probe in estimating the end-to-end path capacities in a number of fixed rate wireless network configurations.

Modeled after Li's simulation configurations in [16] , we use the ns-2 simulator with CMU wireless extensions [1]. The wireless channel is tuned to imitate the Lucent WaveLan card at 2Mbps, with the effective transmission range set at 250 meters and the interference range at 550 meters. Nodes remain stationary during the simulations, and all simulated data packets are preceded by an RTS/CTS exchange, in accordance with the 802.11 standards. Adhoc probe is implemented in ns-2 and used to estimate the end-to-end path capacity in various wireless network configurations.

A. Distinguishing One-way and Round-trip Techniques

We first wish to show the difference of one-way AdHoc Probe and round-trip CapProbe techniques by evaluating them on a simple one-hop wireless link (source and destination separated by 200 meters with no external interference). Capacity estimation results from both one-way and round-trip techniques are shown in Fig. 5.

The results are as expected. As earlier explained in our throughput model, the round-trip estimates are always about half of the one-way estimates in the one-hop wireless link scenario. Wireless contention and backoff resulting from packet collisions (between the 2nd packet of the packet pair and the acknowledgement of the 1st packet) is the main reason why round-trip CapProbe consistently measures a lower end-to-end capacity. One intuitive way to see this is that the path capacity is shared by the two directions of the probing flow, and thus it is halved. Since in our applications we are interested in the "one direction" capacity, we will restrict ourselves to AdHoc Probe in the remainder of the paper.

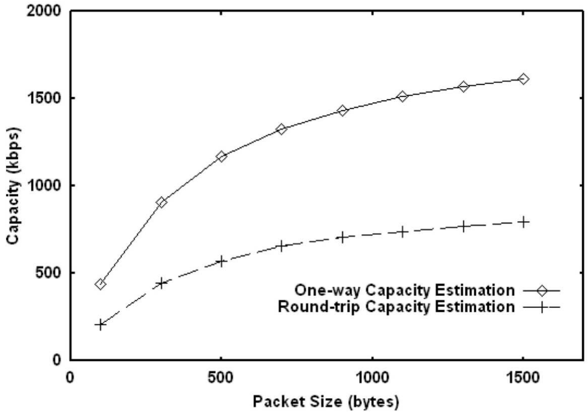

Fig. 5. Capacity estimation results of a wireless link (with no interference from other nodes) using one-way and round-trip CapProbe.

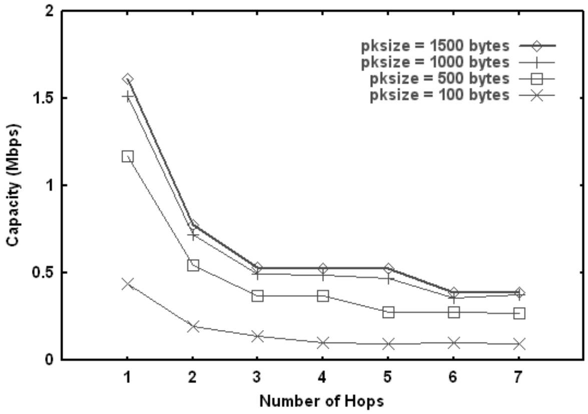

Fig. 6. Capacity estimation along a chain of nodes with different chain lengths and probing packet sizes.

B. Path Capacity on Chain Topologies

This subsection studies the capacity on single chain topologies, where packets originate from the first node and are forwarded to the last node on the chain. Forwarding nodes are expected to contend and interfere with their neighbors, meaning that the effective path capacity will be adversely affected.

Here, we use the same scenario as shown in Fig. 4. The transmission range (marked by solid-line circle) of an 802.11 node is 250 meters, the interference range (marked by a dotted-line circle) is 550 meters, and the nodes are placed on a straight line with 200 meters in between. We have run a set of AdHoc Probe simulations on chain topologies of various packet sizes and path lengths; the results are shown in Fig. 6. As expected, the estimate value increases as the packet size increases, consistent with the analytical results from Eq. 4.

Moreover, the effective end-to-end capacity decreases as the chain grows longer, demonstrating an inverse relationship between the two variables. When the chain length exceeds four, at the packet size of 1,500 bytes, the estimated end-to-end capacity converges to 400∼500 Kbps. It is approximately $\frac{1}{4}$ of the single cell capacity, close to the analytical results of $\frac{1500}{1500+20+14+34} \times 2 \times \frac{1}{4} \approx 478$ Kbps as discussed earlier.

With the same wireless network configuration as specified in [16], AdHoc Probe was able to achieve end-to-end capacity estimation that closely matches the analytical prediction ($\frac{1}{4}$ of single hop capacity). A previous empirical method for evaluating wireless end-to-end capacity [16] converged to a lower estimate of single cell capacity (∼250 Kbps) using the same channel and packet size assumptions employed in our simulation.

That method measured the path rate by pushing a UDP stream and evaluating the achieved throughput. AdHoc Probe attains a path capacity estimate that is more consistent with analytical results as compared with [16], with considerably less intrusion. In general, a stream or flow based testing strategy like the one reported in [16] can be more appealing as it gives a measure of real achieved throughput (i.e., what you measure is what you will actually get this instant with a UDP stream). However, the technique can be too disruptive for some applications. Moreover, the estimate is more limited as it depends on the type of stream used (e.g., the UDP experiment cannot be directly used to predict TCP throughput). If there is other traffic in the network, the probing stream will interact with it in a way that is very difficult to analyze; it will be difficult then to extract the "idle path" rate estimate that we are set out to discover. AdHoc Probe in contrast is much less intrusive. It only needs one "good" sample to correctly estimate the path capacity no matter what the cross load and interference is. Consequently, the path capacity so estimated is of more general use as it can be employed to model and predict the throughput of different types of streams (e.g., UDP, TCP, etc) in different loading conditions.

C. Path capacity within the same interfering domain

Next, we evaluate the capacity of a highly interfered wireless path. More precisely, we wish to validate the C/N relationship derived from the model in Section IV. To this end, we have designed a simulation experiment where the hops of the multihop path are all in the same collision domain. The topology and configurations used here are the same as in subsection V-B, except that the distance between a node and its next-hop neighbor is reduced to 10 meters here. We have run AdHoc Probe using the packet size of 1,500 bytes with various numbers of hops. Fig. 7 shows the average estimation results of 20 runs, as well

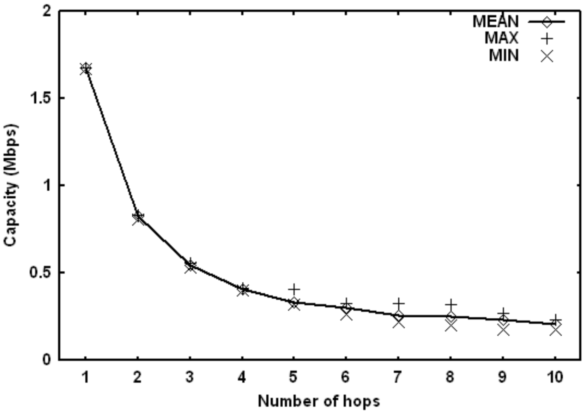

Fig. 7. Capacity estimation of wireless multihop connections within the same collision domain.

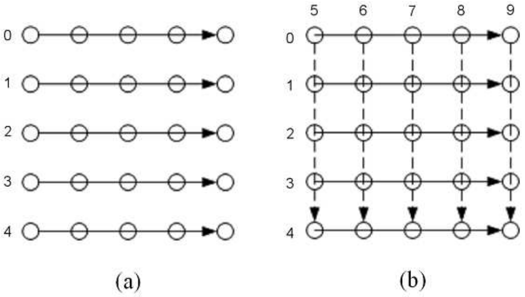

Fig. 8. A regular 5x5 Grid topology. (a) left-to-right horizontal flows; (b) both left-to-right horizontal and top-down vertical flows.

as maximum and minimum estimates, at each number of hops. As predicted by the model, the end-to-end capacity estimate decreases as the inverse of the number of interfering nodes (or equivalently, the number of hops in the same collision domain).

D. Path Capacity Estimation on Grid Topologies

Since grid topologies are more representative of ad hoc configurations than chain topologies, we now consider the n x n regular grid shown in Fig. 8. Nodes are placed 200 meters away from their horizontal and vertical neighbors. Radio transmission range is set to 250 meters, and the radio interfering range is set to 550 meters. We consider two types of traffic patterns: horizontal flows only (Fig. 8-a), and horizontal plus vertical flows (Fig. 8-b). Path capacity is measured for the mid horizontal row via AdHoc Probe; other paths carry flows with Poisson distribution at an average rate that varies up to 100Kbps. Fig. 9 and Fig. 10 show the respective path capacity estimates.

As shown in Fig. 9 and Fig. 10, AdHoc Probe gives the correct estimate for all configurations. For example, in a 4 x 4 regular grid topology, the path length is 4 and (as shown in Fig. 6) capacity is

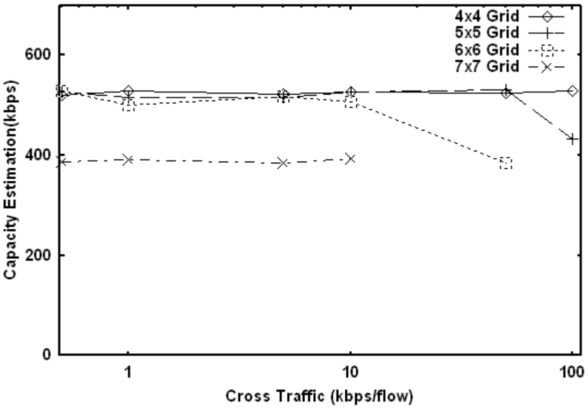

Fig. 9. Capacity estimation in a grid wireless network with horizontal cross traffic.

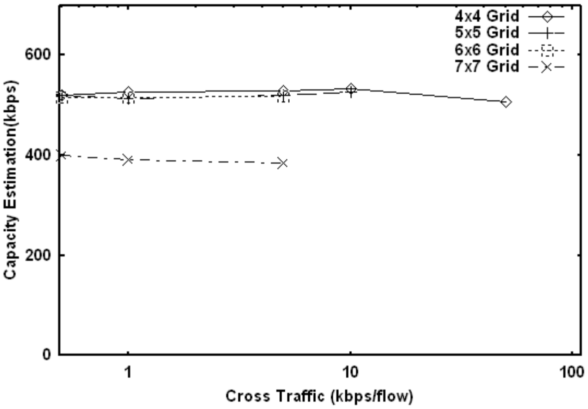

Fig. 10. Capacity estimation in a grid wireless network with both horizontal and vertical cross traffic.

520Kbps. For the 7x7 grid, path length is 7 and capacity is 400Kbps. The estimates become incorrect (by defect) when the grid becomes totally saturated with cross traffic. For example, consider the 4x4 grid with both horizontal and vertical cross traffic. Because of the transmit and interfere ranges, only one packet at a time, vertical or horizontal can go through the grid with perfect scheduling. For 1,500B packet size, this translates into an upperbound on the maximum 4x4 grid capacity of 60 Kbps per flow. Fig. 10 shows that below saturation the 4x4 capacity estimate is accurate. Around saturation, at 60Kbps, the estimate is not correct by a small “defect”. In this situation, the grid never offers an “idle window” through which AdHoc Probe pairs can sneak through and provide min sum estimates. All pairs tend to be separated by an extra amount due to intervening traffic. This leads to underestimates, as clearly shown.

This experiment essentially reconfirms for the ad hoc network case the property that we already discovered in wired networks. Namely, CapProbe (and now Ad Hoc Probe) can estimate the capacity

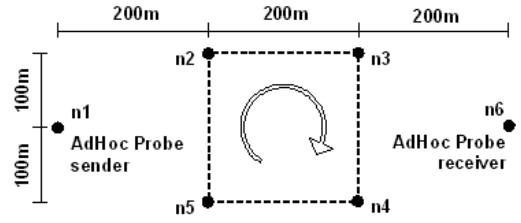

Fig. 11. Scenario of fixed source/destination and mobile intermediate nodes.

correctly up to the point where the path becomes saturated! This is not very intuitive in multihop ad hoc networks where the packets in the pair travel separated by 4 hops. Any cross traffic transmitted by 2-hop neighbors during this 4 hop window will interfere with the pair and invalidate the min sum requirement. Thus, for the same network loading, the risk of interference with the packet pair appears to be much higher in ad hoc nets than in wired nets. Yet, AdHoc Probe can still estimate the correct capacity!

E. Ad Hoc Probe with mobile nodes/Hosts

After evaluating AdHoc Probe in stationary wireless scenarios, we now apply AdHoc Probe to carefully engineered mobile scenarios where we can control the "path breakage" rate. We want to study AdHoc Probe robustness to path breakage and path reestablishment. Fig. 11 depicts the scenario where the AdHoc Probe sender and receiver are fixed, and the intermediate forwarding hosts are mobile. For each wireless node, radio transmission range is set to 250 meters, and the radio interfering range is set to 550 meters. Node 2, 3, 4, and 5 are moving as a group and clockwise along the dotted square path with a fixed speed (which is varying from 10 m/sec to 100 m/sec). Naturally, similar performance is observed when the intermediate nodes move randomly. But, our scenario permits us to control the path breakage rate. In the scenario, Dynamic Source Routing (DSR) is used so that the AdHoc Probe has a path (3 hops) to destination (but the route continuously breaks and must be reconstructed due to mobility). The capacity estimation is performed with 20 separate runs for each configuration (each run with 200 packet pair samples). The average capacity estimates and standard deviations are shown in Fig. 12.

The results clearly show that AdHoc Probe is able to accurately estimate path capacity. In other words, enough min sum samples are collected while the path is up to allow the correct rate estimation. AdHoc Probe measured around 530 Kbps capacities, which confirmed the estimation results we obtained from the simulation of the chain topology of 3 hops length.

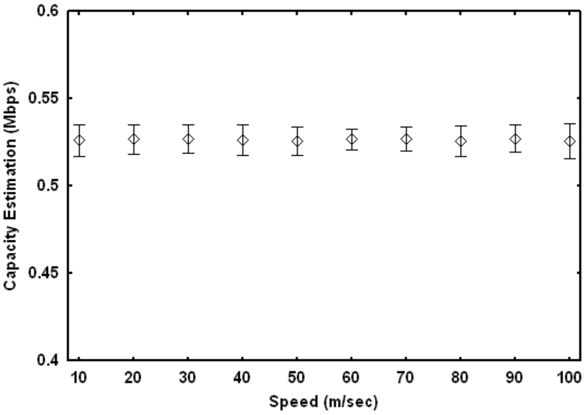

Fig. 12. AdHoc Probe capacity estimates (average of 200 runs and the standard deviation) in simulation scenario.

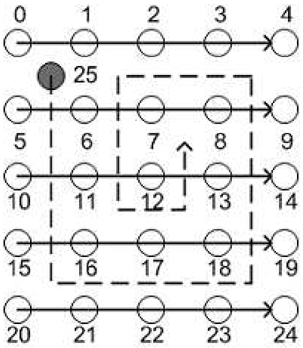

Fig. 13. MANET Scenario, where node 25 (The Host) is moving along the dotted path with a fixed speed (1m/sec).

F. Monitoring path capacity in Ad Hoc Networks w/ Mobile End Hosts

After the mobile intermediate nodes scenario, we now examine another mobile scenario, where one of the AdHoc Probe hosts is mobile and the intermediate hosts are fixed.

Fig. 13 depicts the mobile Host scenario, consisting of 25 stationary nodes (numbered from 0 to 24) and one mobile Host (node 25). Stationary nodes are arranged as a 5x5 grid, spaced 200 meters apart from their horizontal and vertical neighbors. The mobile node travels along the indicated path at a fixed speed of 1 meter/second.

For the purpose of reducing the number of variables, every node is configured to transmit at a fixed rate of 2Mbps, with a transmission range of 250 meters and an interfering range of 550 meters. DSR routing protocol is used.

AdHoc Probe is deployed to measure the capacity of the connection from a fixed source node (node 0) to the mobile Host (node 25). In this simulation, AdHoc Probe uses packet size = 1500 bytes and

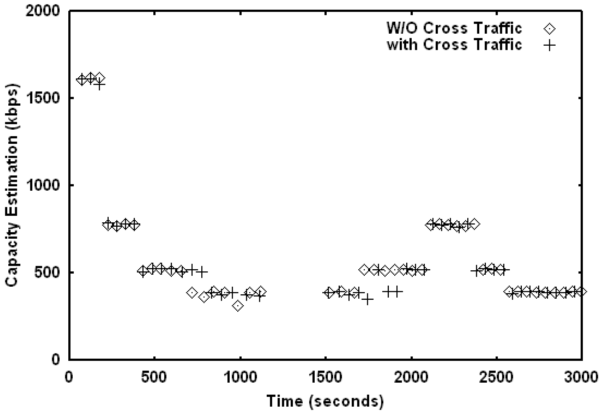

Fig. 14. AdHoc Probe capacity estimates (average of 200 runs and the standard deviation) in simulation scenario.

probing frequency = 4 samples per second, resulting in a rate equal to 1500*8*2*4 = 96 Kbps. The path capacity is then estimated with and without cross traffic. In the cross traffic experiment, five Poisson UDP flows, each at an average rate of 5kbps, are set up from nodes 0, 5, 10, 15, 20 to nodes 4, 9, 14, 19, 24, respectively. Results are collected and depicted as points in Fig. 14.

In Fig. 14, we report the capacity estimation. Note that the capacity will vary as the node moves since the path length in hops changes. AdHoc Probe correctly estimates the capacity as a function of hop distance regardless of cross traffic. The results in Fig. 14 match the results of the chain topology in zero load reported in Fig. 6.

When node 25 moves away from its initial position (100,100) towards its first destination (700,700), the estimated end-to-end capacity decreases sharply from the original 1600kps to 780kbps, and then to 500kbps. This is because the hop count increases from one to four during the same period.

VI. Capacity estimation with Auto Rate Modems

In this section, we study the performance of AdHoc Probe with auto rate modems. We first present an overview of auto rate techniques in subsection VI-A, followed by simulation results of AdHoc Probe in such an environment in subsection VI-B.

A. *Overview of Auto Rate techniques*

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 [21].

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 [11], RBAR [8], and AARF [13]) and *Opportunistic Scheduling* schemes (e.g., OAR [22] and MAD [10]).

Adaptive Rate schemes could be either sender based or receiver based. Auto Rate Fallback (ARF) [11] 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 [13], Lacage 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.

Receiver Based Auto Rate (RBAR) [8] is, as the name implies, a receiver based algorithm aiming at optimizing the application throughput. In RBAR, the sender embeds the ongoing transmission rate information in the RTS packet. Upon receiving the RTS packet, the receiver calculates the transmission rate to be used based on SNR and an a priori wireless channel model. The calculated rate is sent back to the sender in the CTS packet. The sender then transmits its DATA frames using the specified rate. Since the RTS/CTS exchange occurs just before the transmission of the DATA packet, RBAR is able to adequately adjust the data rate to the varying channel conditions, and it is generally accepted as the rate adaptation scheme of choice. However, since RBAR modifies the standard RTS/CTS packets, it is difficult to deploy it in existing 802.11 networks.

Besides adapting the transmission data rate, Opportunistic Scheduling schemes transmit multiple data frames without RTS/CTS exchanges when the wireless channel is in good condition. Since the MAC layer overhead is reduced, the effective capacity is increased. Opportunistic Auto Rate (OAR) [22] is the first

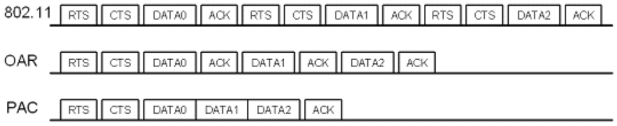

Fig. 15. Illustration of 802.11, OAR, and PAC schemes.

opportunistic scheduling scheme, which transmits multiple packets (by treating them as fragments) when channel condition allows a higher data rate. Later, Ji et al proposed a Medium Access Diversity (MAD) framework to actively exploit time and space channel dynamics at the MAC layer [10]. Not only can OAR be incorporated into the MAD framework, it can also work with the Packet Concatenation (PAC) scheme to further eliminate the ACK packets among multiple data packets in OAR. Fig. 15 illustrates the MAC layer interactions of the original 802.11 scheme, OAR scheme, and the PAC scheme. Again, since OAR and PAC need to modify 802.11 RTS/CTS and DATA/ACK exchanges, they are not compatible with the existing 802.11 standard.

AdHoc Probe will work with any of the above schemes and will provide a stable path rate estimate if the adaptation scheme reaches equilibrium. In the next subsection, we evaluate AdHoc Probe in Auto Rate wireless networks using NS-2 simulator with the auto rate extension described in [2]. The extension uses the Ricean fading model [20], and it supports both RBAR and OAR. In the simulation, the transmission data rate of 802.11b MAC is adapted among 2, 5.5 and 11Mbps.

B. Tracking Capacity Changes in Auto Rate Environment

To force RBAR and OAR to adjust the rate, we set up a simulation experiment with a two node network where destination B moves away from source A at the speed of 1 meter/sec. Four AdHoc Probe samples are injected per second, and a capacity estimate is computed every 200 packet pair samples. The same simulation scenario is applied to both RBAR and OAR schemes.

The results depicted in Fig. 16 show the relationship between the estimated A, B link capacity (with RBAR and OAR respectively) and the distance from the source to the destination. After 700 seconds, when the destination B is 150 meters away from source A, the estimated capacity under RBAR and OAR drops to 4Mbps and 5Mbps, respectively. When the destination is 350 meters away from the source during time 1500 to 2200 seconds, the estimated capacity of RBAR and OAR schemes dropped to around 1.5Mbps. Let us reemphasize the fact that RBAR and OAR adapt the sending rate to the wireless channel S/N ratio.

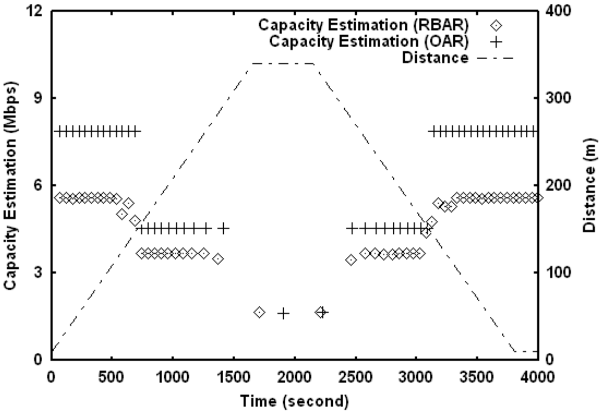

Fig. 16. Simulation results of AdHoc Probe on an auto rate wireless link with different displacements.

Finally, the distance between A and B decreases again during the interval 2200 to 4000 seconds. RBAR and OAR raise the sending rate since the channel conditions have now improved.

OAR achieves a higher capacity than RBAR for the same distance because OAR sends multiple data frames without additional RTS/CTS exchanges. It is remarkable that AdHoc Probe can capture this difference and correctly report the improvement introduced by OAR.

The simulation results in this section have amply confirmed the relationship between the source-destination distance and the path capacity on multi-rate devices. We defer the discussion on interference-triggered rate adaptation to the testbed experiment section.

VII. Testbed Experiments

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

A. 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 $\sim$ 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

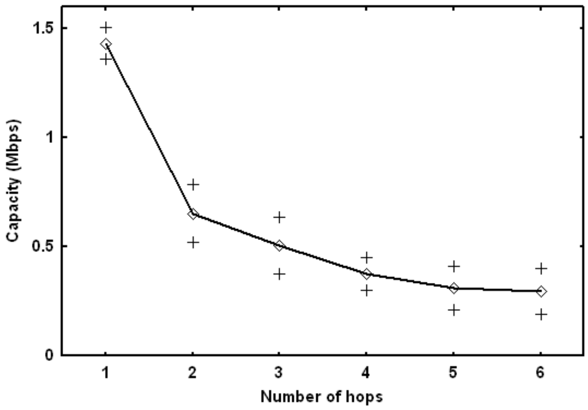

Fig. 17. Experiment results of AdHoc Probe on wireless multihop testbed (transmission rate is 2Mbps on each link).

is conducted without cross traffic, and the average and standard deviation of the capacity estimates is presented below in Fig. 17.

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. The results confirm what we learned in our simulations.

B. 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 Fig. 18.

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 about 3Mbps, corresponding to 5.5 Mbps modem rate. When the distance between source-destination reaches 100 meters, the average effective

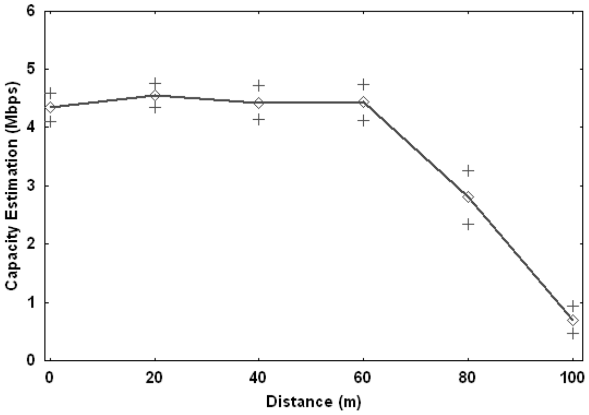

Fig. 18. Experiment results of 802.11b one hop connection (auto rate) with different distance between two hosts.

capacity of about 1Mbps, which again, corresponds to 1Mbps modem rate[3]. The experimental results thus confirm the relationship between source-destination distance and path capacity as discussed in subsection VI-B.

C. Experiment 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.

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. Fig. 19 illustrates the testbed configuration. Two 802.11b laptops (i.e., AdHoc Probe sender and receiver) are placed 10 meters apart, and two Bluetooth laptops (using Bluetooth v1.1 protocol stack) 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 (1,500 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 in an attempt to adapt to the changing channel conditions.

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 Fig. 20.

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

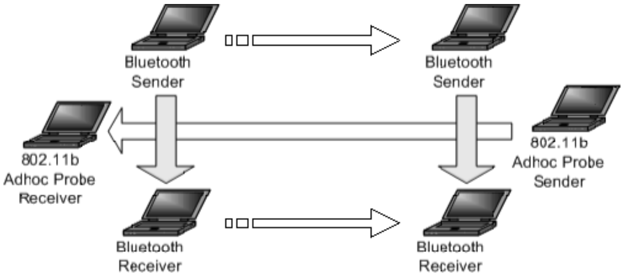

Fig. 19. Auto Rate 802.11b Testbed with Bluetooth interference.

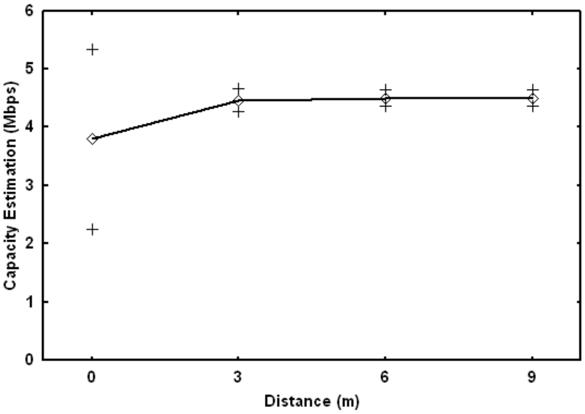

Fig. 20. Auto Rate 802.11b Testbed with Bluetooth interference.

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.

D. Remote estimation of ad hoc network capacity from the wired Internet

In the last experiment, we estimate the capacity of a path that starts from the wired Internet and terminates in the ad hoc network. This type of measurement is important in "opportunistic" ad hoc network applications where for example a server must deliver a multimedia file to a mobile user currently roaming in an ad hoc network connected to the Internet (e.g., an urban vehicular network). The testbed configuration employed in this experiment is the same as in the fixed rate multihop experiment as shown in subsection VII-A, except that here the probing packets are sent from the Internet host (i.e., on a wired path), to the access point (which is a laptop with both wired and wireless interfaces), and from the access point via the wireless multihop path to the destination. Note that the procedure is still end-to-end, the

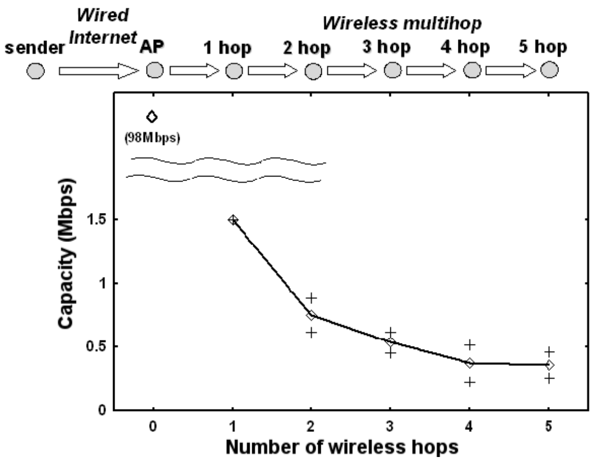

Fig. 21. Experiment results of estimating capacity from Internet to ad hoc networks.

packet pair interval is measured by the destination and the path capacity estimate is returned to the Internet source. Fig. 21 shows the experiment results.

From the results, AdHoc Probe measures 98Mbps capacity on the wired segment (i.e., when the end point is the access point), which is consistent with 100Mbps fast Ethernet bottleneck. When the end point is wireless, the bottleneck shifts to the ad hoc network and AdHoc Probe measures path capacity consistent with the results in subsection VII-A. As expected, AdHoc Probe functions well on both wired and wireless paths and combinations thereof. It is thus an appropriate tool for remotely estimating path capacity of wireless ad hoc extensions of the Internet.

VIII. Conclusions

In this paper, we have reviewed the definition of "path capacity" in ad hoc wireless networks and have proposed a technique - AdHoc Probe - that can efficiently measure such capacity. The technique is a packet-pair based technique inspired to CapProbe, the equivalent tool used in the Internet. AdHoc Probe measures the wireless path capacity that the user would achieve in absence of competing traffic. The procedure is totally end-to-end and is thus independent of the specific protocols implemented in the ad hoc network. In this paper we have presented analysis, simulation, and experimental testbed results. We have validated AdHoc Probe in fixed rate wireless networks with varying path lengths (in hops). We have also showed that AdHoc Probe works well in a loaded network until the network itself becomes completely congested. We evaluated AdHoc Probe in auto rate wireless networks by varying the displacements and as well as the wireless interference. The results showed that AdHoc Probe is able to accurately measure path capacity in all cases of fixed rate networks. Moreover, AdHoc Probe is able to track the rate adaptation of an auto rate wireless link timely and correctly. Finally, AdHoc probe was applied across the Internet to measure the path capacity to a remote wireless network. In summary, AdHoc Probe has provided accurate measurements in all possible environments. It is simple, timely, accurate, and much less intrusive than some previously proposed techniques based on sending entire test streams. Work is now underway to extend the AdHoc Probe concept to probe other important properties of wireless ad hoc networks such as path load and random loss rate.

ACKNOWLEDGEMENT

We are grateful to the editors and anonymous reviewers for their insightful comments. This material is based upon work supported by the National Science Council under grant number NSC 94-2218-E-001-002 and the National Science Foundation under grant number CNS-0435515.

REFERENCES

[1] Network simulator (ns-2). http://www.mash.cs.berkeley.edu/ns/.

[2] Oar. http://ww.ece.rice.edu/networks/software/OAR/OAR.html.

[3] A. Balk, M. Gerla, M. Sanadidi, and D. Maggiorini. Adaptive video streaming: Pre-encoded mpeg-4 with bandwidth scaling. *Elsevier Computer Networks*, 44:415–439, March 2004.

[4] E. M. Belding-Royer and C. E. Perkins. Evolution and future directions of the ad hoc on-demand distance vector routing protocol. *Ad Hoc Networks Journal*, 1:125–150, July 2003.

[5] L.-J. Chen, G. Yang, T. Sun, M. Y. Sanadidi, and M. Gerla. Enhancing qos support for vertical handoffs using implicit/explicit handoff notification. In *ICST QShine*, 2005.

[6] C. Dovrolis, P. Ramanathan, and D. Moore. What do packet dispersion techniques measure? In *IEEE Infocom*, 2001.

[7] M. E. Dyer. Linear algorithms for two- and three- variable linear programs. *SIAM Journal on Computing*, 13:31–45, 1983.

[8] G. Holland, N. Vaidya, and P. Bahl. A rate-adaptive mac protocol for multi-hop wireless networks. In *ACM MobiCom*, 2001.

[9] V. Jacobson. Pathchar: A tool to infer characteristics of internet paths. ftp://ftp.ee.lbl.gov/pathchar/.

[10] Z. Ji, Y. Yang, J. Zhou, M. Takai, and R. Bagrodia. Exploiting medium access diversity in rate adaptive wireless lans. In *ACM MobiCom*, 2004.

[11] A. Kamerman and L. Monteban. Wavelan ii: A high-performance wireless lan for the unlicensed band. *Bell Lab Technical Journal*, Summer:118–133, 1997.

[12] 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.

[13] M. Lacage, M. H. Manshaei, and T. Turletti. Ieee 802.11 rate adaptation: A practical approach. In *ACM MSWiM*, 2004.

[14] K. Lai and M. Baker. Measuring bandwidth. In *IEEE Infocom*, pages 235–245, 1999.

[15] K. Lakshminarayanan, V. N. Padmanabhan, and J. Padhye. Bandwidth estimation in broadband access networks. In *IMC*, 2004.

[16] J. Li, C. Blake, D. Couto, H. I. Lee, and R. Morris. Capacity of ad hoc wireless networks. In *ACM MobiCom*, 2001.

[17] D. L. Mills. Network time protocol specification, implementation and analysis. Technical report, IETF RFC 1305, March 1992.

[18] S. B. Moon, P. Skelly, and D. Towsley. Estimation and removal of clock skew from network delay measurements. In *IEEE Infocom*, 1999.

[19] V. Paxson. On calibrating measurements of packet transit times. In *ACM SIGMETRICS*, 1998.

[20] R. J. Punnoose, P. V. Nilutin, and D. D. Stancil. Efficient simulation of ricean fading within a packet simulator. In *IEEE VTC-Fall*, 2000.

[21] D. Qiao, S. Choi, A. Jain, and K. G. Shin. Miser: An optimal low-energy transmission strategy for ieee 802.11a/h. In *ACM MobiCom*, 2003.

[22] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly. Opportunistic media access for multirate ad hoc networks. In *ACM MobiCom*, 2002.

[23] K. Xu, M. Gerla, and S. Bae. How effective is the ieee 802.11 rts/cts handshake in ad hoc networks? In *IEEE Globecom*, 2002.

[24] K. Xu, X. Hong, and M. Gerla. An ad hoc network with mobile backbones. In *IEEE ICC*, 2002.

[25] L. Zhang, Z. Liu, and C. H. Xia. Clock synchronization algorithms for network measurements. In *IEEE Infocom*, 2002.

APPENDIX: CLOCK DRIFT CALIBRATION

When the clock drift is skewed, the time offset of the two machines is not constant. We use $\triangle(t)$ 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

$$\begin{aligned} S_i &= (T_{recv1,i} - T_{send,i} - \triangle(T_{recv1,i})) - \\ &\quad (T_{recv2,i} - T_{send,i} - \triangle(T_{recv2,i})) \\ &= S'_i - \triangle(T_{recv1,i}) - \triangle(T_{recv2,i}) \end{aligned} \tag{5}$$

Unlike the result in the previous subsections, $S'_k = \min_{i=1\dots n} S'_i$ does not infer $S_k = \min_{i=1\dots n} S_i$. 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 $\triangle(t)$.

The calibration problem has been first studied in [19], 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 [18] and [19]. However, it only works well if the network delays are normally distributed. [18] also formulates this problem as a linear program and solves it using standard algorithms in [7]. In this study, we employ

Algorithm 1 Algorithm for determining the trend of offset function, $\triangle(t)$.

```
$trend \leftarrow 0$;
for $i = 2$ to $N$ do
    if $S'_i > S'_{i-1}$ then
        $trend \leftarrow trend + 1$;
    else if $S'_i < S'_{i-1}$ then
        $trend \leftarrow trend - 1$;
    end if
end for
if $trend > K$ then
    Report $\triangle(t)$ is with increasing trend;
else if $trend < K$ then
    Report $\triangle(t)$ is with decreasing trend;
end if
```

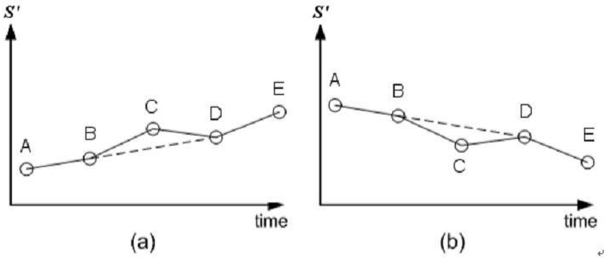

Fig. 22. Illustration of convex-hull based calibration approach: (a) the increasing trend case; (b) the decreasing trend case.

a convex-hull based approach [25], 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 $\triangle(t)$ is monotonic with either an increasing or a decreasing trend. The trend of $\triangle(t)$ can be determined by the *alg_trend* algorithm, as shown in Alg. 1 with a manual threshold setting K. The smaller K is, the more sensitively the algorithm behaves.

Once the trend of $\triangle(t)$ is found, the convex-hull calibration algorithm is applied. As shown in Fig. 22, the calibration algorithm [25] will find a lower bound of skewed measurements when the trend is increasing, or an upper bound when the trend is decreasing. We assume $\triangle(t)$ is a piecewise linear function; therefore, it is sufficient, but not necessary, that all good samples lie on the lower/upper bound curve.

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