# Access Link Capacity Monitoring with TFRC Probe

Ling-Jyh Chen, Tony Sun, Dan Xu, M. Y. Sanadidi, Mario Gerla

Department of Computer Science, University of California at Los Angeles
Los Angeles, CA 90095, USA

***Abstract*-- Accurate estimation of network characteristics based on end to end measurements is an important and challenging problem. With the increasing variety and sophistication of Internet access methods, the mobility of the users and the need for seamless handoff across them, monitoring access link capacity becomes critical for efficient multimedia delivery. In fact, this allows the multimedia server to properly adjust its sending rate to the rapidly changing access speed of the mobile client. This problem is challenging because most of the existing capacity monitoring schemes are active, like Pathrate (and therefore introduce extra overhead on the bandwidth limited access link). Moreover, the access link is often asymmetric (eg, ADSL, 1xRTT, etc) thus preventing the use of round trip monitoring schemes such as Pathchar and CapProbe. In this study, we propose and evaluate a passive, one-way link capacity monitoring tool called TFRC Probe. With TFRC Probe the source can monitor the forward direction capacity of both asymmetric and symmetric access links, and can rapidly and accurately adapt its transmissions rate accordingly. We validate TFRC Probe with both simulation and testbed experiments, and show that TFRC Probe is a very flexible tool that can accurately track frequent changes in access capacity.***

***Index terms*—capacity estimation, CapProbe, passive estimation, TFRC Probe**

## I. Introduction

Knowledge of link capacity is particularly important for network management, pricing, and QoS support, especially in emerging technologies such as in overlay, P2P, sensor and grid networks. With the emerging complexity of wireless network connection technologies, the link capacity of a network connection may vary dramatically due a variety of factors, such as vertical handoff (e.g. a handoff between 802.11b and 1xRTT technology), dynamic channel allocation (e.g. 1xRTT and GPRS), and wireless channel quality (e.g. 802.11b). For these sophisticated settings, knowing the link capacity will permit the source to rapidly and accurately adapt the outbound data transmissions rate. Therefore, it is of increasing interest nowadays to achieve an accurate and "on-line" monitoring of link capacity.

The basic idea of using monitoring probes to estimate link capacity can be achieved through either active or passive measurement. Active measurement is a common approach that injects measurement (probe) packets into the network. The obvious drawbacks of active measurement have motivated the development of passive measurement techniques, which aimed to detect network properties without disturbing on-going network services (via traffic already flowing through the network). Monitoring probes are also required to reflect changes in link capacity (e.g. a handoff between 802.11b and 1xRTT technology) in a timely manner, so a system can adjust its data transmission properties based the most "up-to-date" information. Moreover, to enable seamless network protocols functionalities and other management operations, monitoring probes would have to maintain end-to-end properties. Summarizing the design goals discussed above, it is apparent that an ideal network monitoring technique should a) provide correct information, b) work passively without adding excess overhead to the networks, c) promptly react to occurrences in network events and d) maintain end-to-end semantics.

Addressing all of the above requirements for capacity estimation and monitoring, we propose TFRC Probe, an on-line capacity monitoring technique achieved through embedding the CapProbe algorithm [7] within the TFRC protocol [4]. Different from the *round-trip* nature of CapProbe algorithm, we specifically designed TFRC Probe to monitor the link capacity of the *forward* direction link only. This is based on the realization that capacity information on the forward direction link conveys critical information for any data transferring operations involving asymmetric links. Since information traffic on asymmetric links such ADSL are usually ten times more intensive on forward direction link (download) as oppose to reverse direction link (upload) (e.g. video streaming and file downloading), the ability to appropriately establish the upper-bound of servers' sending rate can provide much assistance in regulating the quality/speed/smoothness of data delivery services. For instance, a previous study has shown that TFRC is slow in responding to a drastic capacity increase [5], and has indicated that a fast rate adaptation algorithm can significantly improve multimedia delivery (for example, by adjusting source rate, content and format) [1]. In this study, TFRC Probe is validated with both simulations and testbed experiments, and has been proven to be quite an effective tool in providing accurate capacity estimation and monitoring.

The rest of the paper is organized as follows. In section 2, we present work related to this study. In section 3, we briefly describe TFRC protocol and detailed the concepts of TFRC Probe. In section 4, we evaluate the accuracy of

TFRC probe in estimating link capacity through series of NS2 simulations. In section 5, we present results from our testbed experiments to validate the monitoring capability of TFRC Probe in Internet and wireless networks. Section 6 concludes the paper.

## II. Background and Related Work

Earlier approaches for estimating link capacity relied on either delay variations among probe packets as in pathchar [6], or dispersion among probe packets as in Nettimer [9] and Pathrate [2]. Other tools such as pchar and clink [3] use variations of the same idea as pathchar. Pchar also uses regression to determine the slope of the minimum RTT versus the probing packet size. In [9], Lai and Bakers propose kernel density techniques to statistically filter packet pair dispersions. In [12], Paxson showed that the dispersion distribution can be multi-modal (e.g. in multi-channel links), and proposed the use of Packet Bunch Modes, a technique involving packet trains of different lengths. Dovrolis' [1] analysis clearly revealed that the dispersions distribution 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.

CapProbe [7] is a recently proposed capacity estimation technique, which has been shown to be fast and accurate over a large range of scenarios. CapProbe is based on a simple and fundamental observation that both expanded and compressed probing samples must have experienced queuing delays on either the first or the second packet of the probing packet pair. By observing the sum of the delays from a packet pair sample, called the *delay sum*, a "good" packet pair sample can be easily identified as the minimum of all the delay sums. The dispersion of the packet pair sample with the minimum delay sum can then be used to estimate the correct link capacity.

The simplicity of CapProbe algorithm allows full integration of its technique with myriads of other data transmission protocols with minimal modifications. With the only requirement being the inclusion of back-to-back transmitted packets to the protocol data traffic. The first implementation of CapProbe uses ICMP messages since every ICMP request is replied to immediately. However, such "out of band" capacity estimation traffic is often considered inappropriate for mobile computing environments, especially when the link capacity is fairly small. In this paper, we will discuss a passive link capacity estimation technique by embedding CapProbe techniques within TRFC protocol. The details of the design and protocol integration will be presented in the following subsections, and the evaluation of the proposed integration will be discussed in the simulation and experiments sections.

## III. Approach

We will briefly review the underlying operating principles of TFRC protocol in section A, which is followed by a detailed description of a passive capacity estimation technique dubbed TFRC Probe in section B. Design tradeoff between capacity resolution and probing frequency of this estimation technique is also addressed.

### *A. TFRC Overview*

TCP-Friendly Rate Control (TFRC) is an equation based unicast multimedia streaming protocol proposed in [4]. TFRC mimics the TCP long-term throughput by utilizing the response function below to control the upper bound of sending rate [4]:

$$T = \frac{s}{R\sqrt{\frac{2p}{3}} + t_{RTO}\left(3\sqrt{\frac{3p}{8}}\right)p\left(1+32p^2\right)} \qquad (1)$$

$T$ represents the upper bound of the sending rate, which is determined by packet size $s$, round trip time $R$, loss event rate $p$, and the TCP retransmission timeout value $t_{RTO}$.

TFRC is designed to facilitate flow controlled TCP friendly transport of data streams without strict error controls. It is designed to increases the sending rate gradually over time. For example, the maximum increase of TRFC sending rate is capped at just 0.14 packets per RTT, or 0.22 packets per RTT with history discounting as described in [4]. Furthermore, TFRC is also designed to respond smoothly to data loss events, instead of cutting down the sending rate drastically upon every single loss event.

In order to achieve smoother data transmission in TFRC, the sender and the receiver are required to cooperate with each other. The sender is responsible for computing the smoothed round-trip time $R$ using an exponentially weighted moving average, and determining the retransmit timeout value $t_{RTO}$. The sender is also responsible for adjusting its sending rate $T_{actual}$ to be close to $T$, which is derived from the equation.

On the other end, the receiver is responsible for calculating the loss event rate $p$ and sending the information back to the sender once per round-trip time. The loss event rate is obtained by maintaining an array of the last eight loss intervals. This loss interval array is continuously updated and a weighted average of the loss intervals is computed. The reported loss event rate $p$ is defined as the inverse of the weighted average.

### B. TFRC Probe

TFRC probe is a passive capacity monitoring technique realized through embedding the CapProbe algorithm within the original TFRC protocol. TFRC Probe is designed to meet the following objectives: a) accurate capacity estimation, b) fast estimation process c) minimal traffic overhead and modification to the original TFRC protocol. Like CapProbe, TFRC Probe estimates link capacity, since such information is important for the streaming server to adjust its sending rate and media quality. However, as a difference from CapProbe, which estimates However, as a difference from CapProbe, which estimates the *round-trip* capacity of the path, TFRC Probe is designed to estimate the *one-way* link capacity on the forward direction link. The one-way estimation is important when the link is asymmetric. . In the following we will describe how TRFC can accomplish those objectives.

#### 1) Accurate Capacity Estimation

Capacity estimation in the *one-way* fashion has attracted increasing amount of research interest of late. This is due to the fact that most bandwidth consumption (e.g. data streaming services, file download, etc) occurs on the forward direction link. As a result, capacity information on the forward direction link is helpful for the data servers to properly adjust its sending rate or data quality, whereas the traditional *round-trip* estimation may not be adequately representative on asymmetric links (e.g. ADSL and satellite links).

Following the fundamental concepts of CapProbe, passive capacity estimation capability can be added to TFRC by simply sending a portion of data packets back-to-back and estimating the link capacity based on the measured dispersion and end-to-end delay of these back to back packets [7]. Figure 1 compares the difference between TFRC and TFRC Probe. In the original TFRC, as illustrated in Figure 1-a, transmission of data packets is paced and evenly distributed, based on the computed sending rate. This is beneficial to multimedia streaming applications which require a smooth and stable sending rate. For the purpose of CapProbe-based capacity estimation, however, paced transmission lacks the packet pairs that are crucial to the scheme. In order to perform passive capacity estimation following the CapProbe algorithm, a modification is made in TFRC Probe such that after every $n^{th}$ data packet is sent out, the TFRC Probe sender immediately transmits the next data packet without waiting for the pacing interval. In other words, TFRC Probe creates a back-to-back sampling packet pair for every $n$ packet. The default value of $n$ is set to *20* in our experiments. Figure 1-b highlights the differences between the two schemes.

Additionally, in order to achieve *one-way* capacity estimation, the back-to-back sampling packets are time-stamped with the sending time ($T_0$) of the corresponding packet pair. Upon receiving the sampling packets, TFRC Probe receiver measures the *one-way delay* of each packet in the packet pair ($T_1$ and $T_2$) by subtracting $T_0$ from its respective receiving time. The dispersion ($T_2$ - $T_1$) and *delay sum* ($T_2$ + $T_1$) are then calculated, and the capacity estimation is made by following the CapProbe algorithm [7]. The capacity estimation results can be reported to the TFRC Probe sender using either the original ACK packets or "out-of-band" reporting packets. In our implementation, the capacity estimation results are carried by the regular ACK packets; therefore, no additional traffic overhead is introduced.

Note that, the measured one-way delay ($T_1$ and $T_2$) may not exactly represent the experienced transmission time of each sampling packet, since the sender and the receiver hosts may not be correctly synchronized in advance. However, the dispersion measurement and the process of finding the sampling packet pair with the minimum delay sum are still effective in TFRC Probe, as long as the frequency of the time ticks are the same on the two end hosts. Since the CapProbe algorithm only relies on the

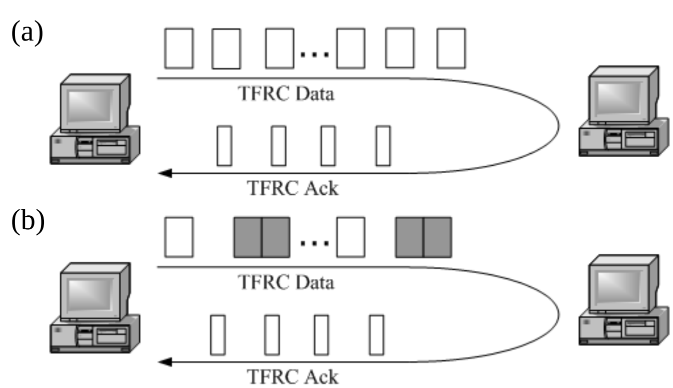


**Figure 1: (a) original TFRC (b) TFRC Probe (the gray ones are back-to-back sampling packets)**

dispersion of the sampling packet pair which has the minimal delay sum, the capacity estimation can be thus accurate even when the two end systems are not properly synchronized.

### *2) Fast Link Capacity Estimation*

Besides providing accurate capacity estimation, a successful link capacity monitoring tool should promptly “capture” each capacity change event when it occurs (e.g. adaptation of transmission modes on the 802.11b link, or a vertical handoff between two different connecting technologies). It turns out that the capacity estimation process needs to be fast and the estimate needs to be updated frequently. The speed of capacity estimation is determined by two factors, namely the *convergence speed* and the *sampling rate*. Though TFRC Probe inherits the outstanding convergence speed from CapProbe algorithm, the speed of capacity estimation in TFRC Probe still relies on the sending rate of the probing packets. More specifically, suppose $R_{send}$ is the sending rate of data packets, $S$ is the number of samples needed to get a reliable capacity estimation, $P$ is the data packet size, $t$ is the expected time to get one capacity estimation, the relation of these properties can be represented in Eq. 2 and Eq. 3.

$$R_{send} \times t = n \times S \times P \times 8 \quad (2)$$

$$\Rightarrow P = \frac{R_{send} \times t}{8 \times n \times S} \quad (3)$$

In our implementation, the default setting of $n$ is 20 (the probing packets are sent every 20 data packet), and $S$ is set to 20 (the capacity estimation results are reported every 20 samples). Since $R_{send}$ is maintained by TFRC rate control algorithm, the optimal data packet size $P$ is then obtained by Eq. 3 given the expected time for each capacity estimation update, $t$, which is set to 5 seconds by default. By iteratively update $P$ according to the sending rate $R_{send}$ for every $S$ samples, TFRC Probe is able to monitor the link capacity with resolution of $t$ seconds. Wisely tuning the resolution factor $t$, TFRC Probe is able to be notified the capacity changes while monitoring the network links.

## IV. SIMULATION

In this section, the monitoring ability of TFRC Probe is verified by a variety of configurations in NS-2 simulator [10]. A set of simulations are performed to evaluate the accuracy and speed of the capacity estimation, and different types of cross traffic are used to simulate different network dynamics. The topology we used in the simulation is depicted in Figure 2, where bottleneck link (between node 3 and 4) is shared by all the data flows and configured as an asymmetric link with various capacities in the *forward* direction and fixed capacity (100Kbps) in the *backward* direction. TFRC Probe and CapProbe are independently performed on the path from node *1* to node *6*, and the cross traffic (if it exists) is generated from node 7 to 10, 8 to 9, 11 to 14, and 12 to 13 respectively.

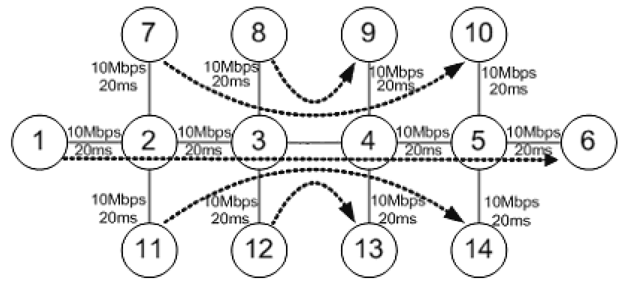

**Figure 2: Simulation Scenario**

**Table 1: Types of cross traffic**

| Cross Traffic | Description |
|---|---|
| Type I | 4 FTP flows (from node 7 to 10, 8 to 9, 11 to 14, and 12 to 13); 1500 bytes/packet |
| Type II | 4 CBR flows (from node 7 to 10, 8 to 9, 11 to 14, and 12 to 13); 500 bytes/packet; 80% load on the bottleneck |
| Type III | 16 Pareto flows with alpha = 1.9 (4 flows from node 7 to 10, 4 flows from 8 to 9, 4 flows from 11 to 14, and 4 flows from 12 to 13); 1000 bytes/packet; 80% load on the bottleneck |

Three different types of cross traffic, detailed in Table 1, were used to examine the speed and the accuracy of our probing techniques under various network conditions. Cross traffic type I and II are simple FTP/CBR connections between multiple senders and receivers. For cross traffic type III, multiple Pareto flows were used to model after web traffic [13] . Different capacity values are assigned on the link from node 3 to node 4 to create the bottleneck link. In each experiment, we also collect the estimated capacity after 20 packet samples and again using 50 packet samples to evaluate the speed of the estimation. We summarized the results in Table 2.

From the results, TFRC Probe shows high accuracy in all the test cases, regardless of different types of cross traffic. It is also evident that accurate estimation can be achieved with merely 20 packet samples for all of the scenarios. While CapProbe measures the round-trip

**Table 2: TRFC Probe-Capacity Estimation Result**

| | | TFRC Probe | CapProbe | TFRC Probe | CapProbe | TFRC Probe | CapProbe | TFRC Probe | CapProbe |
|---|---|---|---|---|---|---|---|---|---|
| no cross traffic | 20 samples | 100 K | 100 K | 500 K | 100 K | 1 M | 100 K | 5 M | 100 K |
| | 50 samples | 100 K | 100 K | 500 K | 100 K | 1 M | 100 K | 5 M | 100 K |
| cross traffic type I | 20 samples | 100 K | 100 K | 500 K | 100 K | 1 M | 100 K | 5 M | 100 K |
| | 50 samples | 100 K | 100 K | 500 K | 100 K | 1 M | 100 K | 5 M | 100 K |
| cross traffic type II | 20 samples | 100 K | 100 K | 500 K | 100 K | 1 M | 100 K | 5 M | 100 K |
| | 50 samples | 100 K | 100 K | 500 K | 100 K | 1 M | 100 K | 5 M | 100 K |
| cross traffic type III | 20 samples | 100 K | 100 K | 500 K | 100 K | 1 M | 100 K | 5 M | 100 K |
| | 50 samples | 100 K | 100 K | 500 K | 100 K | 1 M | 100 K | 5 M | 100 K |
| bottleneck capacity of the *backward* direction link | | 100 Kbps | | 100 Kbps | | 100 Kbps | | 100 Kbps | |
| bottleneck capacity of the *forward* direction link | | 100 Kbps | | 500 Kbps | | 1 Mbps | | 5 Mbps | |

capacity of a link, TRFC probe is specialized to measure the forward link capacity. Again, it should be emphasized that capacity information conveyed by the forward link is critical for most data transferring protocols (e.g. video streaming and file downloading) in setting the appropriate upper-bound for their sending rates. This is especially true for protocol operating on asymmetric links such as ADSL. The simulation results indicate that the proposed TFRC Probe approach is beneficial for such purposes.

## V. EXPERIMENTS

In this section, we present our testbed experiment results to validate the correctness and capabilities of TFRC probe. Measurement result on links of various speed and networking technologies is addressed in section A; the effectiveness TRFC probe in monitoring wireless capacity is discussed in section B.

### A. *Experimental Verification of TFRC Probe*

The first set of experiment evaluates the accuracy of TFRC probe on wired links of different connecting technology, which is followed by a set experiment on wireless scenarios. For these experiments, the adaptive packet size feature is temporarily disabled to allow validation of TFRC probe functionality. The link connections tested were: 100Mbps Ethernet, 2M/128Kbps DSL, 802.11b (11, 5.5, 2, and 1 Mbps transmission modes), and 1xRTT (150Kbps). The DSL DownLink is estimated by hosting the TFRC Probe sender on the Internet and the receiver behind the DSL link; whereas the DSL UpLink is estimated by hosting the TFRC Probe sender behind the DSL link and the receiver on the Internet. The probing packet size is fixed at 1000 bytes for each experiment, which is repeated five times for each link. Table 3 and Table 4 shows the experiment results (estimated capacity and required time for completion) after collecting 20, 50, and 100 samples.

From the results, TFRC Probe were able to estimates the capacity accurately and rather rapidly (in terms of number of samples). Specifically, TFRC Probe is able to measure the bottleneck capacity within 20 samples and within 10% of the actual values in almost all experiments. Note that the effective capacity of 802.11b is usually smaller than the physical channel capacity due to the MAC layer overhead (e.g. RTS/CTS packets and CSMA/CA mechanisms). Compared with the CapProbe results reported in [7], TFRC Probe measures the capacity of the forward direction link; whereas CapProbe always measures the UpLink capacity of the DSL link since it relies on the "round-trip" based estimation.

The results also show that the speed of capacity estimation is highly related to the link capacity. For instance, while estimating capacity of the 100Mbps Ethernet link, it takes only 5.5 seconds to obtain 20 samples in the experiments; whereas, while it took 70 seconds to estimate the 128Kbps DSL UpLink. It is obviously desirable to speed up the estimation process in order to perform efficient "monitoring" task and to provide "up-to-date" information. The packet size adaptation feature is thus necessary, and its effect will be evaluated in the next subsection.

### B. *Capacity Monitoring of Wireless Links*

Figure 3 depicts the experiment results of 802.11b capacity monitoring. The experiment testbed is created on a Linux based system, and the 802.11b transmission mode is manually changed with '*iwconfig*' command.

**Table 3: Capacity estimation of TFRC Probe on the wired links of different technologies**

| | | Run 1 | | Run 2 | | Run 3 | | Run 4 | | Run 5 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Capacity | Time | Capacity | Time | Capacity | Time | Capacity | Time | Capacity | Time |
| Ethernet 100Mbps | 20 samples | 92.18 | 5.5 | 94.92 | 5.5 | 94.25 | 5.6 | 107.99 | 5.5 | 94.07 | 5.6 |
| | 50 samples | 98.17 | 5.8 | 104.31 | 5.7 | 94.25 | 5.9 | 94.02 | 5.7 | 94.07 | 5.8 |
| | 100 samples | 98.17 | 6.1 | 104.31 | 6.0 | 94.25 | 6.2 | 94.02 | 6.1 | 94.07 | 6.2 |
| DSL DownLink 2Mbps | 20 samples | 1.982 | 7.0 | 1.807 | 8.6 | 1.710 | 8.8 | 1.782 | 6.7 | 1.835 | 6.7 |
| | 50 samples | 1.982 | 12.7 | 1.865 | 12.4 | 1.835 | 12.6 | 2.145 | 12.6 | 1.652 | 10.5 |
| | 100 samples | 1.982 | 20.9 | 1.865 | 18.9 | 1.835 | 18.9 | 2.145 | 19.0 | 1.652 | 16.9 |
| DSL UpLink 128Kbps | 20 samples | 0.114 | 70.9 | 0.122 | 69.5 | 0.115 | 72.7 | 0.110 | 81.5 | 0.119 | 71.4 |
| | 50 samples | 0.114 | 145.0 | 0.122 | 137.6 | 0.115 | 146.0 | 0.115 | 147.5 | 0.112 | 142.3 |
| | 100 samples | 0.115 | 265.6 | 0.122 | 258.5 | 0.119 | 258.6 | 0.115 | 266.0 | 0.112 | 257.5 |

Capacity: Mbps, Time: second

**Table 4: Capacity estimation of TFRC Probe on the wireless links of different technologies**

| | | Run 1 | | Run 2 | | Run 3 | | Run 4 | | Run 5 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Capacity | Time | Capacity | Time | Capacity | Time | Capacity | Time | Capacity | Time |
| 1xRTT 150Kbps | 20 samples | 0.186 | 115.9 | 0.187 | 87.1 | 0.186 | 68.3 | 0.140 | 116.0 | 0.186 | 61.6 |
| | 50 samples | 0.140 | 156.4 | 0.140 | 122.3 | 0.187 | 113.2 | 0.140 | 188.5 | 0.140 | 113.1 |
| | 100 samples | 0.140 | 230.0 | 0.140 | 182.9 | 0.140 | 187.0 | 0.140 | 286.8 | 0.140 | 212.3 |
| 802.11b 1Mbps | 20 samples | 0.65 | 17.0 | 0.91 | 17.8 | 0.85 | 18.7 | 0.94 | 17.7 | 0.92 | 17.9 |
| | 50 samples | 0.93 | 25.6 | 0.93 | 27.8 | 0.91 | 28.2 | 0.94 | 26.3 | 0.93 | 27.8 |
| | 100 samples | 0.91 | 39.6 | 0.93 | 41.6 | 0.91 | 42.6 | 0.92 | 41.2 | 0.93 | 42.3 |
| 802.11b 2Mbps | 20 samples | 1.48 | 16.7 | 1.80 | 16.8 | 1.81 | 16.9 | 1.77 | 17.9 | 1.80 | 18.0 |
| | 50 samples | 1.48 | 21.8 | 1.74 | 21.2 | 1.68 | 21.6 | 1.09 | 23.9 | 1.69 | 23.5 |
| | 100 samples | 1.49 | 29.1 | 1.47 | 29.6 | 1.80 | 29.8 | 1.72 | 32.3 | 1.73 | 32.5 |
| 802.11b 5.5Mbps | 20 samples | 3.73 | 14.8 | 3.86 | 15.3 | 4.36 | 14.9 | 3.83 | 15.1 | 4.45 | 14.6 |
| | 50 samples | 4.18 | 18.4 | 3.86 | 19.7 | 3.80 | 19.5 | 4.06 | 19.3 | 3.87 | 18.6 |
| | 100 samples | 4.18 | 23.2 | 3.92 | 24.4 | 3.80 | 24.0 | 4.06 | 24.0 | 3.87 | 22.8 |
| 802.11b 11Mbps | 20 samples | 8.09 | 7.6 | 7.60 | 7.5 | 7.91 | 7.5 | 7.00 | 7.6 | 6.26 | 7.9 |
| | 50 samples | 8.09 | 9.0 | 7.60 | 10.1 | 6.23 | 9.6 | 7.00 | 10.2 | 6.26 | 10.2 |
| | 100 samples | 7.84 | 12.7 | 7.60 | 11.8 | 6.23 | 11.6 | 6.60 | 12.0 | 6.26 | 11.9 |

Capacity: Mbps, Time: second

During the experiment the transmission mode is set to 2, 5.5, 11, 5.5, and 2 Mbps mode at times {50, 100, 150, 200, 250second}. The packet size adaptation feature of TFRC Probe is enabled in the experiments; thus, the packet size is now a function of the TFRC Probe sending rate, number of data packets between two samples (*n*, which is set to 20), number of samples for each estimation (*S*, which is set to 20), and expected time for each estimation (*t*, which is set to 5 second) according to Eq. 3.

The estimated capacities illustrated by Figure 3 are very consistent and closely matches the effective 802.11b capacities for most of the reported values. There are some inaccurate results reported from the experiment, several over estimation of capacity were reported at around $250^{th}$ second, and under estimation of capacity were reported around $205^{th}$ second. These estimation inaccuracies are likely influenced by the dynamic nature of the wireless channel. The estimation accuracy can be improved by simply increase the number of required samples for each estimation (*S*); however, it might decrease the estimation speed in accordance to Eq. 2.

Figure 3 also shows the adaptation of TFRC Probe packet size, as expressed in Eq. 3. It is evident that the packet size of TFRC probe increases when the sending rate increases and decreases when the sending rate decreases. While operating at 5.5 and 11 Mbps modes, the packet size remained stalled due to the maximum packet size setting (1500 bytes) in our implementation.

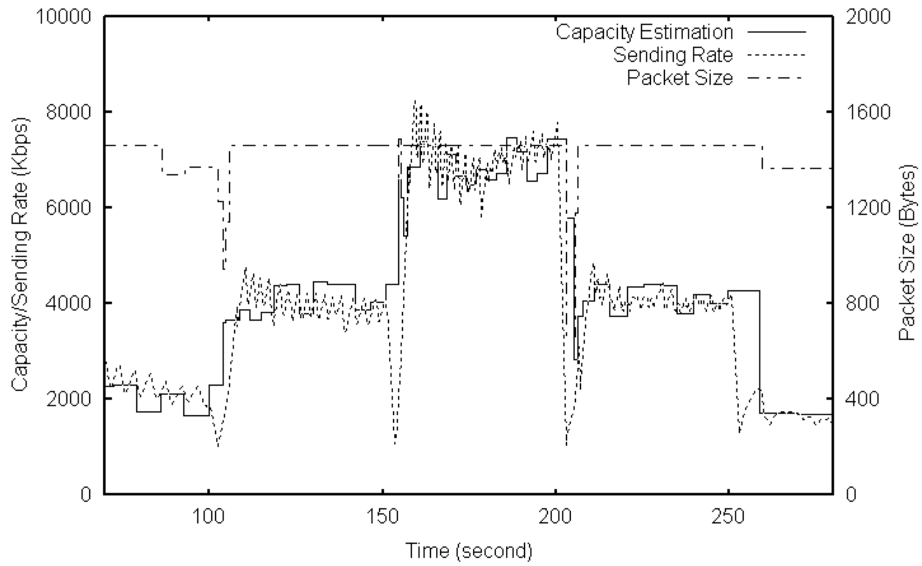


**Figure 3: Capacity monitoring of 802.11b connection**

## VI. CONCLUSION

In this study, we present TFRC probe, an efficient passive monitoring tool that provides accurate capacity estimation. TFRC Probe combines CapProbe and TFRC, achieving faster capacity estimation with lower overhead. Moreover, TFRC Probe maintains its probing rate (and therefore accuracy) constant by adjusting the probing packet size in accordance to TFRC sending rate. This guarantees "up-to-date" capacity information.

Validated with both simulations and various testbed experiments, TFRC Probe has demonstrated high accuracy and convergence speed. From those results, we conclude that TFRC Probe is adequate for monitoring network link capacity. It's worth mentioning that the concept of TFRC Probe is not restricted to the TFRC protocol only. The passive capacity estimation concept can be transparently applied to other UDP based application protocols (e.g. RAP, RTP, UDP based FTP and P2P file downloading) and other emerging data transmission protocols (e.g. DCCP [8]).

## REFERENCES


[1] L.-J. Chen, A. Nandan, G. Yang, M.Y. Sanadidi, and M. Gerla. "A Study of Passive Capacity Estimation based on CapProbe," UCLA CSD Technical Report TR040021.

[2] C. Dovrolis, P. Ramanathan, and D. Moore. "What do packet dispersion techniques measure?" in Proceedings of IEEE Infocom'01, 2001.

[3] A. B. Downey, "Using Pathchar to Estimate Internet Link Characteristics," In Proc. of ACM SIGCOMM 1999.

[4] S. Floyd, M. Handley, J. Padhye, and J. Widmer. "Equation-Based Congestion Control for Unicast Applications," In Proc. of ACM SIGCOMM 2000.

[5] A. Gurtov and J. Korhonen, "Measurement and Analysis of TCP-Friendly Rate Control for Vertical Handovers," to appear in ACM MCCR, http://www.cs.helsinki.fi/u/gurtov/papers/vho.html

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

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

[8] E. Kohler, M. Handley, and S. Floyd. "Data Congestion Control Protocol," IETF Internet-Draft, draft-ietf-dccp-spec-06.txt.

[9] K. Lai and M. Baker, "Measuring Bandwidth", In Proceedings of IEEE INFOCOM '99, p. 235-245.

[10] Network Simulator (NS-2), http://www.mash.cs.berkeley.edu/ns/

[11] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. "Modeling TCP Throughput: A Simple Model and its Empirical Validation," In Proc. of SIGCOMM, 1998.

[12] V. Paxson, "Measurements and Dynamics of End-to-End Internet Dynamics", Ph.D. thesis, Computer Science Division, Univ. Calif. Berkeley, April 1997.

[13] Mark Stemm, and Randy H. Katz. "Vertical Handoffs in Wireless Overlay Networks," ACM Mobile Networking (MONET), 1998.

[14] M.S. Taqqu, W. Willinger, R. Sherman, "Proof of a fundamental result in self-similar traffic modeling," SIGCOMM Computer Communications Review, 27: 5-23, 1997.