WBest: a Bandwidth Estimation Tool for IEEE 802.11 Wireless Networks

Mingzhe Li, Mark Claypool and Robert Kinicki
{lmz,claypool,rek}@cs.wpi.edu
Computer Science Department at Worcester Polytechnic Institute
Worcester, MA, 01609, USA

***Abstract*—Bandwidth estimation techniques seek to provide an accurate estimation of available bandwidth such that network applications can adjust their behavior accordingly. However, most current techniques were designed for wired networks and produce relatively inaccurate results and long convergence times on wireless networks where capacity can vary dramatically. This paper presents a new Wireless Bandwidth estimation tool, WBest, designed for fast, non-intrusive, accurate estimation of available bandwidth in IEEE 802.11 networks. WBest is a two-stage algorithm: 1) a packet pair technique estimates the effective capacity over a flow path where the last hop is a wireless LAN (WLAN); and 2) a packet train technique estimates achievable throughput to infer the available bandwidth. WBest parameters are optimized given the tradeoffs of accuracy, intrusiveness and convergence time. The advantage of WBest stems from avoiding a search algorithm to detect the available bandwidth by statistically detecting the available fraction of the effective capacity to mitigate estimation delay and the impact of random wireless channel errors. WBest is implemented and evaluated on an 802.11 wireless testbed. Comparisons with other available bandwidth estimation tools shows WBest to have higher accuracy, lower intrusiveness and faster convergence times. Thus, WBest demonstrates the potential for improving the performance of applications that need bandwidth estimation, such as multimedia streaming, on wireless networks.**

I. Introduction

Due to the shared nature of wireless network communication and MAC layer mechanisms such as wireless layer retries and dynamic rate adaptation, bandwidth estimation is far more challenging when the underlying network includes a wireless Local Area Network (WLAN). Fluctuating wireless channel conditions cause variability in wireless capacity and available bandwidth. Other wireless factors such as reception signal strength and bit error rates (BER) due to path loss, fading and interference and limit the effective bandwidth over a wireless link. While providing satisfying results on wired networks, current bandwidth estimation tools have been shown [1]–[4] to be adversely impacted by IEEE 802.11 wireless network conditions.

Tools that only estimate capacity are not useful for Internet applications such as multimedia streaming that adjust the sending rate in response to other concurrent flows. Video streaming flows require available bandwidth estimates with fast convergence times to avoid client-side buffer underflows and to satisfy users waiting to use the application. Moreover, the inherent variability of a wireless channel implies multiple available bandwidth invocations within a single application stream and this adds minimal intrusiveness as a requirement for a wireless bandwidth estimation tool.

Most of the early bandwidth estimation techniques seek to provide accurate bandwidth information for wired networks at the cost of long convergence times and high intrusiveness. *Self-loading* techniques, such as *Train of Packet Pairs (TOPP)* [5], *pathload* [6] and *pathChirp* [7], probe the end-to-end network path using multiple traffic rates. When the probing rate exceeds the available bandwidth, the probing packets become queued at the tight link[1] router, which results in increased delay on the receiver side. By analyzing the packet delay at the receiver, the available bandwidth at the tight link is obtained from the probing rate when the queuing delay starts increasing. The changing of the probing rate can be managed in different ways. For example, *pathload* uses binary search to adjust the probing rate, *TOPP* uses a linearly increasing probing rate, while *pathChirp* uses an exponentially increasing probing rate. *Probe Gap Model* (PGM) techniques, such as *Initial Gap Increase/Packet Transmission Rate (IGI/PTR)* [9] and *Spruce* [10], measure available bandwidth by estimating the crossing traffic at the tight link and by monitoring the gap changes after the packets pass through the tight link router.

Recent research includes bandwidth estimation techniques specific to wireless networks [1], [3], [11]–[13]. *EXACT* [1] and *IdleGap* [11] both assume RTS/CTS is always enabled and provide only ns-2 simulation results. *CapProbe* [12] tries to avoid the effects of crossing traffic in only estimating capacity. *ProbeGap* [3] estimates available bandwidth in WLANs indirectly from the idle time fraction using one-way delay samples over the wireless link, but requires third party capacity estimation tools. *DietTOPP* [13] uses a reduced TOPP algorithm with a modified search algorithm to determine available bandwidth in wireless networks. However, none of these wireless schemes address wireless layer dynamic rate adaptation.

Packet dispersion techniques, such as packet pair or packet train probing, measure end-to-end capacity on a network path. Introduced in [14]–[16], packet pair dispersion techniques have been enhanced via tools such as *bprobe/cprobe* [17], *sprobe* [18] and *pathrate* [19], [20]. Dispersion techniques send two or more packets back-to-back into the network. After

[1]The *tight link* and *narrow link*, as defined in [8], refer to the hop with the minimum available bandwidth and minimum capacity, respectively.

traversing the narrow link, the time dispersion between the two packets is linearly related to the narrow link capacity. However, using packet dispersion for capacity estimation is impaired by crossing traffic that interferes with probe packets. Additionally, dynamic rate adaptation on WLANs impedes capacity estimation methods that assume fixed capacity during measurement.

Our previous work in packet dispersion [21] provides a detailed analytic model of packet dispersion behavior in wireless networks under varying conditions. By modeling packet dispersion variance in IEEE 802.11 WLANs, the paper approximates estimated capacity variance in terms of the packet dispersion variance derived by the model. By introducing two metrics suitable for wireless network performance analysis, *effective capacity* and *achievable throughput*, the packet dispersion model lead to the development of the Wireless Bandwidth estimation tool (WBest). WBest employs a two-stage algorithm to determine available bandwidth along a flow path when the last hop is a WLAN. In the first stage, WBest utilizes packet pairs to estimate the WLAN effective capacity. In the second stage, WBest sends a packet train at the effective capacity rate to determine achievable throughput and infer available bandwidth.

This paper introduces and analyzes WBest with respect to tradeoffs in accuracy and convergence time. Thorough evaluation in a wireless testbed shows WBest performs better in terms of accuracy, intrusiveness and convergence time than three currently available bandwidth estimation tools: *IGI/PTR*, *pathChirp* and *pathload*. The paper is organized as follows: Section II discusses the WBest algorithm and related issues; Section III describes the experimental setup; Section IV analyzes the experimental results; and Section V provides conclusions and presents possible future work.

II. WBEST ALGORITHM

The WBest algorithm estimates available bandwidth on a network path where the last hop is over a wireless network.

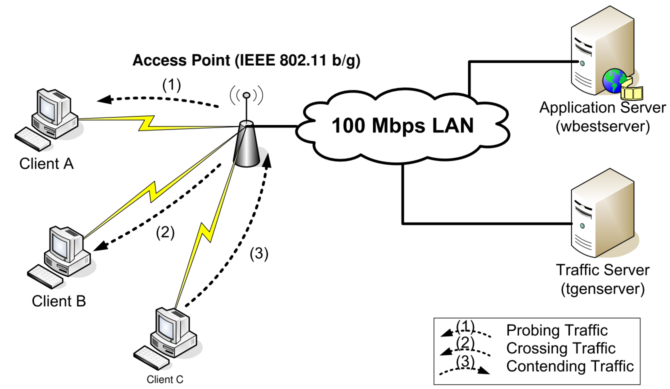

Fig. 1. Network Path with Last Hop Wireless Network.

Figure 1 shows a typical network environment where an application server with a wired Internet connection sends traffic along the network path to a client with a last hop wireless connection. To perform media scaling and buffer optimization for a multimedia stream, the server needs to know the capacity and available bandwidth on the flow path. In Figure 1, network traffic is categorized as probing, crossing and contending. *Probing traffic* (1) is sent by bandwidth estimation tools along the network path through the AP to the client. Wireless channel conditions and other traffic affect probing traffic behavior and produce capacity estimation errors. *Crossing traffic* (2) shares the bottleneck in the direction coming from the AP to associated clients. *Contending traffic* (3) accesses the shared wireless channel and competes with probing traffic on the path of interest. Contending traffic comes from clients to the same AP or from other clients and APs within interference range (known as neighboring AP co-channel interference).

Since available bandwidth is defined as the maximum amount of capacity that a newly arriving flow can acquire at the bottleneck link without negatively impacting existing flows, wireless contending traffic impacts not only capacity sharing at the bottleneck AP, but also reduces the AP available bandwidth due to wireless channel access contention. Hence, while WLAN capacity estimation techniques have to avoid estimation errors due to crossing and contending traffic, techniques to determine WLAN available bandwidth must lower their estimate to account for the reduction in available bandwidth due to both crossing and contending traffic.

A. Assumptions

To make the WBest bandwidth estimation algorithm tractable, the following assumptions are made.

1) Assume the last hop wireless network is the *bottleneck link* on the network path. As the *bottleneck link*, the last hop WLAN has both the smallest available bandwidth (tight link) and the smallest capacity (narrow link) along the network path. Namely, we have the relationship:

$$A \le C_e \le \min_{i=1,..,h-1}(A_i) \qquad (1)$$

where A and C_e are the available bandwidth and effective capacity of the last hop, respectively, h is the number of hops, and A_i is the available bandwidth and capacity of the ith hop. This assumption implies a packet train sent by the source at rate C_e will arrive at the last hop at the rate of C_e [20]. If this assumption does not hold, e.g. for some home wireless networks with a lower capacity broadband Internet connection, the packet train with sending rate C_e will be dispersed before the last hop and arrive at the last hop with a lower rate than C_e. This will cause a conservative under-estimate of the available bandwidth which is typically a better outcome for most applications than an aggressive, over-estimate.

2) Assume no significant changes in network conditions between the effective capacity measurement stage and the available bandwidth estimate stage of the WBest algorithm. While changes in network conditions due to rate adaptation or mobility will impact the estimation results, given algorithm convergence times of milliseconds, the statistical impact of this variability is assumed to be minimal.

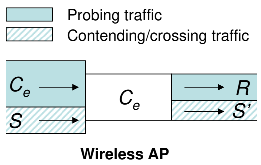

Fig. 2. A Typical Last Hop Wireless Network.

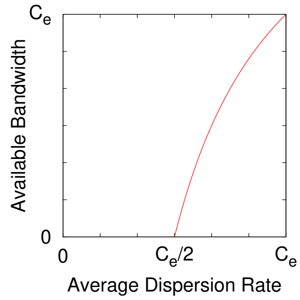

Fig. 3. Estimating Available Bandwidth using Average Dispersion Rate.

3) Assume packet pairs or trains do not overflow any of the router queues along the flow path. A queue overflow at the last hop impacts the accuracy of the estimation results. The possibility of queuing loss is reduced by limiting the number of packet pairs and the length of the packet train sent into the network.

B. Algorithm

Algorithm 1 provides the two-stage WBest algorithm. In the first stage (lines 1-2), n packet pairs are sent to estimate effective capacity, C_e, the maximum capability of the wireless network to deliver network layer traffic [21]. Unlike in wired networks, wireless dynamic rate adaptation alters effective capacity by adjusting the packet transmission rate. Hence, effective capacity is a function of time and packet size:

$$C_e = \frac{\int_{t_0}^{t_1} \frac{L}{T(t)} dt}{t_1 - t_0} \tag{2}$$

where L is the packet size, $T(t)$ is the packet dispersion at time t. To use packet dispersion in a discrete environment, T_i, the ith packet dispersion at time t, is used to represent $T(t)$.

Algorithm 1 WBest Algorithm.

Require: $n > 0$ {Measure effective capacity (C_e)}
1: Send n packet pairs to client
2: $C_e \Leftarrow median(C_i,\ i = 1, .., n)$
Require: $m > 0,\ C_e > 0$ {Measure available bandwidth (A)}
3: Send packet train with length m at rate C_e to client
4: $R \Leftarrow \frac{L}{mean(T_i,\ i=1,..,m)}$
5: **if** $R \geq \frac{C_e}{2}$ **then**
6: $\quad A \Leftarrow C_e \left[2 - \frac{C_e}{R}\right]$
7: **else**
8: $\quad A \Leftarrow 0$
9: **end if**
10: $p \Leftarrow$ packet loss rate in train {Error correction}
11: **if** $p > 0$ **then**
12: $\quad A \Leftarrow A \times (1 - p)$
13: **end if**

While most packet dispersion techniques use the smallest gap between packet pair arrivals to produce a narrow link capacity estimate, WBest uses the median of n packet pair capacity estimates to approximate C_e in the estimation time period and minimize the impact of crossing and contending traffic:

$$C_e = median(C_i),\ i = 1, .., n \tag{3}$$

where C_i is the estimation result of packet pair i and $C_i = \frac{L}{T_i}$. The median is used as opposed to the mean in order to mitigate the impact of outliers in the arrival distribution. In such cases, the mean results in a lower capacity estimate than does the median, and would make the second stage packet train less effective at accurately determining the available bandwidth.

During the second stage of WBest, (lines 3-13), a packet train of length m is sent at rate C_e to estimate available bandwidth. Similar to probe gap techniques, a fluid model is used to estimate the relationship between available bandwidth and dispersion rate. From assumption 1, the arriving rate at the last hop is C_e. Assuming downstream AP traffic can be modeled as a FIFO queue, the downstream probing traffic shares the same ratio of the total amount of traffic before and after the AP queue:

$$\frac{C_e}{C_e + S} = \frac{R}{R + S'} = \frac{R}{C_e} \tag{4}$$

where, as depicted in Figure 2, R is the average dispersion rate at the receiver, S represents the available bandwidth reduction due to last hop crossing and contending traffic and S' is the share of the capacity taking away from probe packets by the crossing and contending traffic. Combining the relationship between available bandwidth and the estimated effective capacity:

$$A = C_e - S \tag{5}$$

and Equation 4, the available bandwidth can be expressed as:

$$A = C_e(2 - \frac{C_e}{R}) = 2C_e - \frac{C_e^2}{R} \tag{6}$$

Note, for a WLAN, *achievable throughput* [21] is R, the average dispersion rate at the receiver for a probing rate of C_e. Using Equation 6, Figure 3 shows the relationship between available bandwidth and achievable throughput. Any achievable throughput less than half of C_e implies zero available bandwidth, and an achievable throughput of C_e implies an idle wireless network.

Packet losses on the wireless network and along the network path impact WBest accuracy. Some tools, e.g. *pathload*, discard estimates when packet losses occur to avoid errors in the estimation computation. However, this yields longer and more variable measurement times. Instead of discarding estimates when packet losses occur, WBest detects packet loss in both packet pairs and packet trains and removes the appropriate pair from the computation. For a packet train, loss rate p is recorded and the available bandwidth estimate reduced (lines 10-13 of Algorithm 1).

WBest's advantages stem from statistically detecting the relative available fraction of effective capacity at the WLAN AP instead of using search algorithms to measure available bandwidth. Many available bandwidth mechanisms detect available bandwidth by measuring the delay changes in the probing traffic. However, random changes in packet delay due to wireless network conditions make it difficult to clearly determine packet delay trends. This reduces accuracy and increases the convergence time, intrusiveness and instability of the estimation scheme. By avoiding a search algorithm to determine the probing rate, WBest is designed to converge faster and yield less estimation error. Instead of probing for the available bandwidth, WBest estimates available bandwidth using the effective capacity. $(2-\frac{C_e}{R})$ in Equation 6 is treated as the fraction of C_e available to all wireless flows. Derived from the ratio of the effective capacity to the average dispersion rate, this available fraction statistically removes random errors while capturing the impact of crossing/contending traffic and rate adaptation inherent in wireless networks.

C. Number of Packet Pairs and Length of Packet Train

The number of packet pairs in the first stage of WBest and the number of packets in the packet train in the second stage play important roles in the accuracy, convergence time and intrusiveness of the algorithm. Generally, more packet pairs and longer packet trains improve accuracy at the cost of higher convergence time and more intrusiveness.

WBest seeks to minimize convergence time and intrusiveness at a given accuracy level. The confidence interval (CI) and the estimated capacity variance from the packet dispersion model [21], σ, are used to determine the required minimum number of packet pairs using:

$$n = \frac{Z^2\sigma^2}{CI^2} \tag{7}$$

where Z is a confidence level constant. For example, assume a streaming video flow wants to bound the effective capacity estimate within 500 Kbps to match the granularity of encoded video scaling levels. To keep the effective capacity estimate within 500 Kbps with 95% confidence, Equation 7 indicates at least 6 (5.34) samples are needed. This is based on $\sigma = 0.59$ Mbps for an 11 Mbps wireless channel and a packet size of 1500 bytes with $Z = 1.96$ and $CI = 500$ Kbps [21]. Similarly, the number of packets m in the packet train can also be computed. With the same available bandwidth estimation bounds and given a modeled maximum $\sigma = 1.38$ Mbps [21] for an 11 Mbps channel and packet size of 1500 bytes with contending traffic, $Z = 1.96$ and $CI = 500$ Kbps, the minimum train size m is 30 (29.26). As real network conditions may change unexpectedly, Equation 7 only approximates the number of samples needed.

The number of packets in a train also impacts the time scale and sensitivity of available bandwidth estimations. In general, the available bandwidth estimate represents the average estimate during the measurement period [8]. As a major part of the convergence time, the time T_m spent to estimate available bandwidth depends on the number of packets m in the train. T_m can be approximated using m and packet size L as $T_m = m * L/C_e$. Furthermore, the probability crossing traffic gets included in the bandwidth estimation is related to the length of the train and packet sizes. Assume CBR crossing traffic is sent at rate S with at least one packet caught by the packet train:

$$\begin{aligned} S * T_m/L &\geq 1 \\ S \geq L/T_m &= C_e/m \end{aligned} \tag{8}$$

The sensitivity of the available bandwidth estimation can be defined based on the number of packets in the train, which has a negative relationship with train length. For instance, to catch crossing traffic sent at rate $C_e/10$, a packet train with at least 10 packets is needed.

Selecting the number of packet pairs and train length is complicated in practice because the bottleneck queue size limits the number of packet pairs and the length of the packet train. The *pathrate* queue size probing method [20] can be used to detect buffer limitations along the flow path. However, this probing method increases intrusiveness and measurement time and is not appropriate for many applications. Since the WBest packet train sending rate is set to the effective capacity of the wireless AP, the probability of queue overflow in the network is determined by the queue size at the AP. Previous research [22] indicates current wireless AP queue lengths range from 40 to 300 packets. Thus, WBest simply limits the packet train to less than 40 packets. To further avoid queue overflow due to packet pairs, WBest inserts a 10 millisecond gap between pairs to reduce the packet pair probing rate during capacity estimation.

As discussed in [1], [12], the ability for crossing and contending traffic to interfere with packet dispersion depends on the relative size of the probe packets versus crossing/contending packets. However, to effectively estimate bandwidth, probing packet size must be close to the packet size of the application using the bandwidth estimator. Thus, using high bitrate video streaming as a motivating example of an application that can benefit from bandwidth estimation, 1500 bytes is used as our probe packet size.

D. Error Detection

Packet loss observed at a wireless receiver may be attributed to either wireless losses or congestion losses (queue overflow). The WBest error correction adjusts for wireless losses. However, while WBest controls the probing traffic sending rate to avoid queue overflow, large amounts of crossing traffic and contending traffic may still produce queue losses that can cause an over-estimate of available bandwidth. In most cases, one can assume that any queuing loss is due to a saturated wireless link with no available bandwidth. However, to guard against queue overflow at an upstream router, Loss Discrimination Algorithms (LDA), such as [23], [24] could be added to WBest to distinguish congestion loss from wireless loss.

Another potential source of estimation error comes from last hop probe packet compression. System factors, such as high CPU load at the wireless clients and user-level timestamps [20] may cause two or more packets to have very close arrival timestamps. Last hop compression can result in recorded arrival rates that are higher than the effective capacity. For example, our measurements show the minimum timestamp from the user level timer is about 2.3 μs. This results in a dispersion rate over 5000 Mbps for a probe packet size of 1500 bytes. Thus, to reduce the error due to last hop compression, when the received timestamp yields a higher rate than the actual sending rate, WBest uses the actual sending rate instead of the dispersion rate to compute available bandwidth.

III. Experiments

WBest is implemented[2] in Linux and evaluated by varying network conditions in an IEEE 802.11 wireless testbed. As shown in Figure 1, the wireless testbed consists of an application server that performs the estimation (wbestserver), a traffic server (tgenserver), a wireless AP and three clients (Client A, B and C). The AP in the testbed is a Cisco Air-AP1121G[3] with IEEE 802.11b/g mode. Both servers are PCs with P4 3.0 GHz CPUs and 512 MBytes RAM and the three clients are PCs with P4 2.8 GHz CPUs with 512 MBytes RAM. All the testbed PCs run SUSE[4] 9.3 with Linux kernel version 2.6.11. The servers connect to the AP with a wired 100 Mbps LAN, and the clients connect to the AP with IEEE 802.11b/g WLAN using Allnet[5] ALL0271 54 Mbps wireless PCI card with a prism GT chipset.[6]

For performance comparison, three popular and available bandwidth estimation tools were selected: *IGI/PTR* v2.0, *pathChirp* v2.4.1 and *pathload* v1.3.2. For the experimental runs, the four tools are run sequentially to estimate the downstream available bandwidth from wbestserver to client A. While all the tools were setup using their default configuration, to provide a fair performance comparison, the following methodology was used to run and summarize the estimation results. Although *IGI/PTR* converges with two results, the PTR results are used as the author suggests. Since *pathload* converges with a range of available bandwidths, the median of the range is used for comparison. During the evaluation, some *pathload* runs never converge under particular wireless channel conditions. These runs were halted if they fail to converge in 100 seconds which is the upper limit of normal convergence time for *pathload*. Since *pathChirp* is designed as a continuous monitoring tool without an explicit convergence policy, convergence follows the author's method described in [7] whereby the difference between the 90th and 10th percentiles of the estimates are computed and convergence is defined when the difference is less than $1/5$ [7] of the available bandwidth (approximately 6 Mbps in our testbed).

To evaluate accuracy, the true available bandwidth of the wireless network under different configurations is needed – referred to here as the *ground truth*. Since determining ground truth during dynamic WLAN conditions is difficult, the ground truth of the available bandwidth is approximated by the downstream throughput of a single saturated CBR UDP flow with a packet size of the Maximum Transmission Unit (MTU) for each case tested. However, due to the elastic nature of the TCP sliding window mechanism, for cases with TCP crossing and contending traffic, ground truth for the available bandwidth in the wireless network is zero. Each evaluation consists of back-to-back runs employing four bandwidth estimation tools and one downstream CBR traffic. For all cases with crossing or contending traffic, the estimations start five seconds after the background traffic starts to let the system stabilize. Similarly, there is a five second delay between the end of one tool and the start of the next to allow background traffic to stabilize.

Table I itemizes the fourteen experimental cases. The base configuration, case 0, has no contending or crossing traffic and no induced changes in wireless conditions. Cases 1-12 include a variety of crossing and contending traffic situations provided by UDP and TCP traffic generators residing on client B, client C and tgenserver. The Multi-Generator Toolset[8] (mgen) v4.2b6 and iperf[9] v2.0.2 are used to generate UDP and TCP traffic, respectively. For case 13, wireless rate adaptation is induced by removing the antenna of a wireless client and reducing the wireless AP's sending power and receiving antenna gain. With a client received signal strength indicator (RSSI) between -70 dbm and -74 dbm, the wireless transmission rate ranged from 1 to 48 Mbps. Figure 4 shows one actual rate adaptation case measured with a wireless sniffer.[10] This rate adaptation case results in 8% wireless layer retries for both the AP and the client.

TABLE I
Evaluation Cases for Experiments.

Case	Crossing Traffic	Contending Traffic
0	None	None
1	Client B: UDP 4.6 Mbps	None
2	None	Client B: UDP 4.6 Mbps
3	Client B: TCP	None
4	None	Client B: TCP
5	Client B: UDP 2.3 Mbps Client C: UDP 2.3 Mbps	None
6	None	Client B: UDP 2.3 Mbps Client C: UDP 2.3 Mbps
7	Client B: TCP Client C: TCP	None
8	None	Client B: TCP Client C: TCP
9	Client B: UDP 2.3 Mbps	Client C: UDP 2.3 Mbps
10	Client B: TCP	Client C: TCP
11	Client B: UDP 2.3 Mbps	Client C: TCP
12	Client B: TCP	Client C: UDP 2.3 Mbps
13	Case 0 with rate adaptation	

Each of the fourteen cases were tested 30 times with

[2]WBest source code can be download from http://perform.wpi.edu/tools
[3]http://www.cisco.com/en/US/products/hw/wireless/ps4570/index.html
[4]http://www.novell.com/linux/
[5]http://www.allnet-usa.com/
[6]http://www.conexant.com/products/entry.jsp?id=885
[7]This ratio is computed from the evaluation setup in [7]
[8]http://pf.itd.nrl.navy.mil/mgen/
[9]http://dast.nlanr.net/Projects/Iperf/
[10]http://perform.wpi.edu/tools/

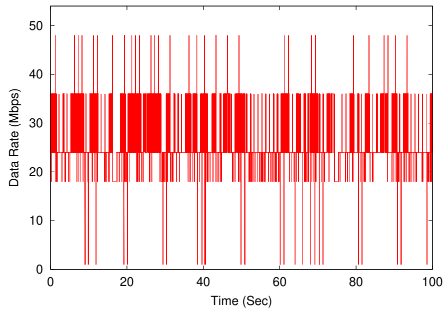

Fig. 4. Rate adaptation Behavior.

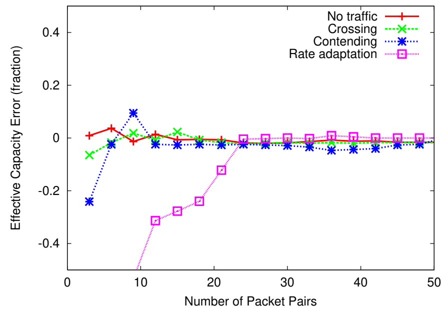

Fig. 5. Analysis of Number of Packet Pairs.

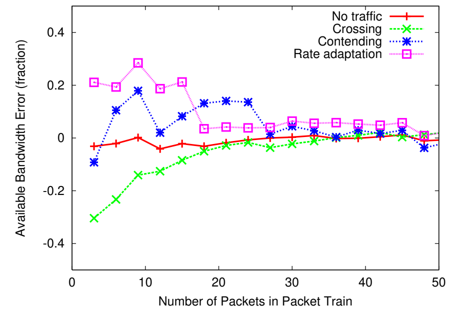

Fig. 6. Analysis of Length of Packet Train.

the median and quartiles reported for all runs. To ensure comparability across different runs, the RSSI range for all wireless clients is between -38 dbm and -42 dbm, and all clients were shown to have the same maximum throughput of about 29 Mbps. To mitigate interference from co-existing campus wireless networks, all experiments were run in our wireless streaming multimedia lab[11] which was painted with an additive[12] to reduce radio transmissions going through the walls. Furthermore, all the experiments were conducted after midnight during the WPI summer break when most campus wireless network is assumed to be in an idle state.

Figure 5 shows the relationship between effective capacity error (modeled in [25]) and the number of packet pairs sent for four typical wireless cases: idle, crossing traffic, contending traffic, and rate adaptation. The real effective capacity is defined as the median of the 90 packet pair run. As the number of packet pairs sent increases, the error decreases. Rate adaptation requires the highest number of packet pairs to produce reasonably accurate measurements. To provide accuracy for all the cases while reducing the impact on the available bandwidth estimations, 30 packet pairs were used in all the WBest evaluations. Similarly, from Figure 6, 30 was chosen as the length of the packet train for all the WBest experiments.

IV. Analysis

For each of the fourteen test cases, Table II lists the median estimated available bandwidth for 30 evaluations runs of each of the four bandwidth estimation tools. The 'ground truth' column gives the available bandwidth measured as CBR UDP throughput with 1500-byte packets or set to zero when a test includes a TCP bulk transfer. For Case 6, UDP traffic from two contending clients yields sufficient WLAN congestion to cause the AP queue to overflow. This mistakenly triggers rate adaptation for the wireless clients. While rate adaptation is a consequence of high wireless contention, the saturated CBR throughput of 9.29 Mbps is not ground truth because higher throughput can be obtained by using a lower offered CBR rate. Thus, ground truth is set to unknown for case 6. For all other cases in Table II, WBest generally provides the most accurate estimation of the available bandwidth compared to the other three bandwidth estimation techniques.

[11] http://perform.wpi.edu/wsml/

[12] http://www.forcefieldwireless.com/defendairadditive.html

Intrusiveness is the total bytes sent by each tool during an estimation and the convergence time is the time spent by each tool to converge to a bandwidth estimation result in each estimation. Table III gives the median of recorded intrusiveness and convergence time over 30 runs for the fourteen tests. WBest yields the lowest intrusiveness and convergence time in every case.

TABLE II
Estimated Available Bandwidth (Median, in Mbps).

#	IGI/PTR	PathChirp	Pathload	WBest	Ground truth
0	8.11	30.15	6.78	28.47	28.94
1	8.74	28.89	6.81	23.24	24.39
2	10.06	27.59	6.91	15.76	20.52
3	1.92	5.00	1.95	1.01	0
4	1.12	14.50	1.69	0.00	0
5	9.99	26.91	7.07	22.87	24.50
6	9.62	26.98	6.78	14.56	-
7	1.48	5.00	1.10	0.00	0
8	0.66	11.97	0.92	0.00	0
9	6.89	25.60	6.47	13.26	16.26
10	0.67	5.72	0.99	0.00	0
11	0.59	9.95	0.48	0.00	0
12	0.77	12.73	1.06	0.00	0
13	5.18	16.79	5.99	13.99	15.26

TABLE III
Intrusiveness (Median, in MBytes) and Convergence Time (Median, in Seconds).

	IGI/PTR		PathChirp		Pathload		WBest	
#	intru	time	intru	time	intru	time	intru	time
0	0.56	1.55	0.45	17.43	1.18	14.88	0.13	0.41
1	0.56	1.42	0.45	17.58	1.55	20.22	0.13	0.42
2	0.47	1.29	0.45	17.62	1.53	17.04	0.13	0.42
3	2.54	17.21	0.46	17.24	1.22	42.06	0.13	0.67
4	1.51	7.86	0.45	17.22	0.86	32.16	0.13	0.44
5	0.56	1.35	0.45	17.68	1.67	19.24	0.13	0.42
6	0.47	1.30	0.45	17.79	1.66	17.33	0.13	0.42
7	3.11	26.69	0.46	18.41	0.95	53.90	0.13	0.70
8	1.98	19.57	0.46	17.89	0.98	55.02	0.13	0.51
9	0.66	1.60	0.45	18.10	1.57	18.42	0.13	0.42
10	2.17	23.30	0.46	17.15	1.24	80.86	0.13	0.98
11	1.79	28.37	0.49	18.27	0.53	30.24	0.13	0.59
12	2.17	15.59	0.46	17.45	1.46	74.94	0.13	0.44
13	0.66	1.86	0.45	17.48	1.66	23.73	0.13	0.42

Due to space limitations, brief analysis is provided for only four cases from the set of fourteen experiments: case 0 - idle channel, case 1 - crossing traffic, case 2 - contending traffic, and case 13 - rate adaptation; [25] provides a complete analysis of all the test results.

When the wireless channel is idle (case 0), available bandwidth and effective capacity are the same. With a measured ground truth throughput of 28.94 Mbps, the available bandwidth/effective capacity is close to the maximum throughput of 31.4 Mbps listed by Cisco.[13] Table II shows that *IGI/PTR* and *pathload* significantly under-estimate available bandwidth. A possible reason is that the probing packet sizes used by these two tools are small – *IGI/PTR* uses a 500 byte packet and *pathload* uses a 200 byte packet. The overhead caused by the sizes of probing packets has been shown to be larger in wireless networks than in wired networks [3], [13]. Hence, the maximum throughput will be lower for these smaller packet sizes. Allowing for reduced maximum throughput of 19.2 Mbps and 11.4 Mbps with 500 byte and 200 byte probe packets, respectively, *IGI/PTR* and *pathload* still significantly under-estimate available bandwidth. *PathChirp* and WBest get an available bandwidth estimate close to the ground truth. However, *pathChirp* tends to overestimate the available bandwidth with a large estimation variance [25]. *Pathload* and *pathChirp* both have long convergence times caused by a search algorithm that adapts the probing rate.

Table III provides intrusiveness and convergence times for case 1 when there is one UDP crossing flow. WBest performs better than the other tools in this case, with low intrusiveness and convergence times and accurate estimated available bandwidth results. The under-estimation caused by the smaller packet sizes used in *IGI/PTR* and *pathload* shows that they are insensitive to crossing traffic, as well. *Pathload*, in particular, has large intrusiveness and convergence times. For case 2 in Table II and Table III, WBest still performs well in the presence of contending traffic. Comparing case 2 with case 0 and 1, *IGI/PTR* and *pathload* are insensitive to contending traffic.

With wireless rate adaptation in case 13, where packet transmission rate and channel access delay vary as in Figure 4, WBest provides the closest available bandwidth estimate below the ground truth. Remember, it is much better for applications such as multimedia streaming to receive an under-estimate of available bandwidth than it is to receive an over-estimate of available bandwidth.

The estimation error of each case is computed [25] and the distributions of error versus the convergence time and error versus intrusiveness are plotted in Figure 7 and 8, respectively. In these figures, on the x-axis, a negative error represents an under-estimation and a positive error represents an over-estimation; and on the y-axis, lower numbers are better. Thus, good, fast estimates lie in the bottom center of the two figures.

IGI/PTR greatly under-estimates available bandwidth with UDP crossing or contending traffic and even when the channel is idle. *IGI/PTR* has widely variable convergence times and intrusiveness, varying by a factor of 20 times for the different

[13]Cisco AVVID Wireless LAN Design, http://www.cisco.com

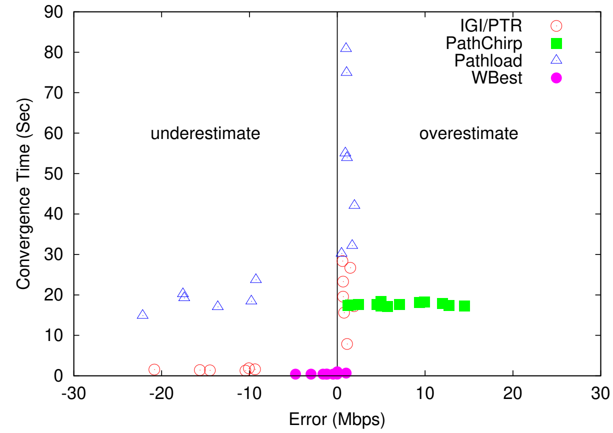

Fig. 7. Summary of All Experiments – Convergence Time versus Error.

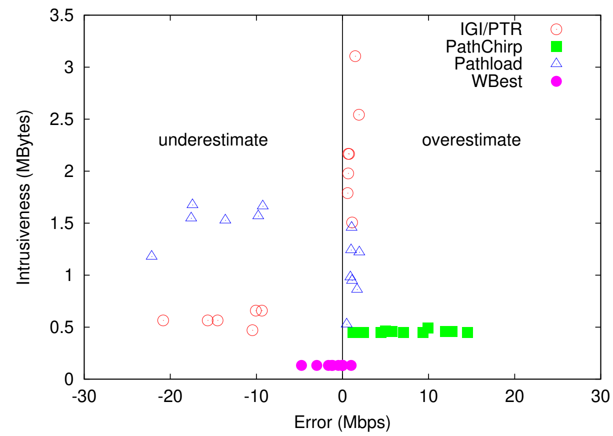

Fig. 8. Summary of All Experiments – Intrusiveness versus Error.

cases. *PathChirp* over-estimates available bandwidth in all cases. *PathChirp* has a consistent convergence time of around 17 seconds and a consistent intrusiveness of about 400 KBytes. *Pathload* greatly under-estimates available bandwidth in most cases including: idle channel, UDP crossing or contending traffic, and rate adaptation. *Pathload* has the longest overall convergence time, taking up to 85 seconds in some cases and fails to converge in 100 seconds for some crossing and contending cases. WBest provides the most accurate estimations compared with these other tools. In most cases, WBest converges in less than half a second with a nearly constant intrusiveness of 130 KBytes.

For wireless networks, *IGI/PTR*, *pathChirp* and *pathload* accuracy is poor because they rely on delay changes to measure available bandwidth. Unfortunately, queuing delay is not the only source of wireless delay. Contention, MAC layer retries and rate adaptation also introduce delay changes to wireless links that disturb the searching algorithms, yield inaccurate results and often increase the convergence times and intrusiveness. Moreover, with higher packet loss rates in WLANs, many estimation techniques discard probes impacted by loss to improve accuracy. This further increases convergence time and intrusiveness.

WBest estimates the available bandwidth without using searching algorithms which means a low, consistent convergence time and intrusiveness. Furthermore, rather than relying on delay measurements to detect the available bandwidth, WBest detects the available bandwidth in terms of fraction of the effective capacity by measuring relative changes in packet dispersion between its two stages. This makes WBest robust even when packet dispersion is affected by wireless conditions.

V. Conclusion

This paper presents WBest, a new bandwidth estimation tool for wireless networks, designed to provide accurate bandwidth estimation without excessively intruding on existing traffic. By utilizing the packet dispersion model [21], WBest avoids depending upon search algorithms to measure available bandwidth. Instead, WBest statistically measures the relative available fraction of the effective capacity, mitigating estimation delay and the effects of wireless channel errors. WBest is compared with other available bandwidth estimation tools in a wireless testbed under a variety of wireless network conditions.

From experiments on a wireless testbed, a few conclusions can be drawn. First, current bandwidth estimation tools are significantly impacted by wireless network conditions, such as contention from other traffic and rate adaptation. This yields inaccurate estimates, high and varying convergence times, and intrusiveness. Thus, current tools are generally impractical for applications such as streaming multimedia that require fast, accurate and non-intrusive bandwidth estimates even when the last hop is over a WLAN. Second, WBest consistently provides fast available bandwidth estimation, with generally more accurate estimates and lower intrusiveness over all conditions evaluated.

Our ongoing work includes applying WBest to multimedia streaming applications to improve media scaling performance and playout buffer optimization in wireless networks. Other possible future work includes the improvement to WBest evaluations under more complex wireless conditions, including experiments that deliberately cause pre-dispersion and pre-compression to validate the WBest model and to enhance WBest robustness during AP queue overflow. A more philosophical future work is to develop and study a new metric to replace available bandwidth when TCP flows are involved. This new metric would involve the ability for a new TCP flow to take its fair share of capacity away from existing crossing and contending TCP flows which the current definition of available bandwidth does not allow.

References

[1] S. Shah, K. Chen, and K. Nahrstedt, "Available Bandwidth Estimation in IEEE 802.11-based Wireless Networks," in *First ISMA/CAIDA Workshop on Bandwidth Estimation (BEst)*, San Diego, CA, USA, Dec. 2003.

[2] Tony Sun, Guang Yang, Ling-Jyh Chen, M. Y. Sanadidi, and Mario Gerla, "A measurement study of path capacity in 802.11b based wireless networks," in *WiTMeMo'05*, Seattle, USA, 2005.

[3] Karthik Lakshminarayanan, Venkata N. Padmanabhan, and Jitendra Padhye, "Bandwidth estimation in broadband access networks.," in *Proceedings of IMC 2004*, Taormina, Sicily, Italy, Oct. 2004.

[4] L. Angrisani, A. Botta, A. Pescape, and M. Vadursi, "Measuring wireless links capacity," *Wireless Pervasive Computing, 2006 1st International Symposium on*, pp. 1–5, Jan. 2006.

[5] Bob Melander, Mats Bjorkman, and Per Gunningberg, "Regression-based available bandwidth measurements," in *SPECTS'02*, July 2002.

[6] Manish Jain and Constantinos Dovrolis, "End-to-end available bandwidth: Measurement methodology, dynamics, and relation with tcp throughput," *IEEE/ACM Transactions in Networking*, , no. 295-308, Aug. 2003.

[7] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell, "pathchirp: Efficient available bandwidth estimation for network paths," in *PAM '03*, La Jolla, CA, USA, Apr. 2003.

[8] R.S. Prasad, M. Murray, C. Dovrolis, and K.C. Claffy, "Bandwidth Estimation: Metrics, Measurement Techniques, and Tools," *IEEE Network*, vol. 17, no. 6, pp. 27 – 35, November-December 2003.

[9] Ningning Hu and Peter Steenkiste, "Evaluation and characterization of available bandwidth probing techniques," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 6, Aug. 2003.

[10] Jacob Strauss, Dina Katabi, and Frans Kaashoek, "A measurement study of available bandwidth estimation tools," in *IMC '03*, 2003, pp. 39–44.

[11] Heung Ki Lee, Varrian Hall, Ki Hwan Yum, Kyoung Ill Kim, and Eun Jung Kim, "Bandwidth Estimation in Wireless LANs for Multimedia Streaming Services," *Adv. MultiMedia*, vol. 2007, no. 1, 2007.

[12] Rohit Kapoor, Ling-Jyh Chen, Li Lao, Mario Gerla, and M. Y. Sanadidi, "Capprobe: a simple and accurate capacity estimation technique," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 67–78, 2004.

[13] Andreas Johnsson, Bob Melander, and Mats Björkman, "Bandwidth measurement in wireless networks," in *Mediterranean Ad Hoc Networking Workshop*, Porquerolles, France, June 2005.

[14] V. Jacobson, "Congestion avoidance and control," in *Proceedings of the ACMSIGCOMM88 Conference*, Stanford, CA, USA, Aug. 1988.

[15] Srinivasan Keshav, "A control-theoretic approach to flow control," in *Proceedings of the ACM SIGCOMM 1991*, Sept. 1991.

[16] Jean-Chrysotome Bolot, "End-to-end packet delay and loss behavior in the internet," in *SIGCOMM '93*, San Francisco, CA, USA, Sept. 1993.

[17] Robert L. Carter and Mark E. Crovella, "Measuring bottleneck link speed in packet-switched networks," *Performance Evaluation*, vol. 27, no. 8, pp. 297–318, Oct. 1996.

[18] Stefan Saroiu, P. Krishna Gummadi, and Steven D. Gribble, "Sprobe: A fast technique for measuring bottleneck bandwidth in uncooperative environments," Aug. 2001, Online: http://sprobe.cs.washington.edu/.

[19] Constantinos Dovrolis, Parmesh Ramanathan, and David Moore, "What do packet dispersion techniques measure?," in *Proceedings of IEEE INFOCOM 2001*, Anchorage, Alaska, USA, Apr. 2001, pp. 905–914.

[20] Constantinos Dovrolis, Parameswaran Ramanathan, and David Moore, "Packet-dispersion techniques and a capacity-estimation methodology," *IEEE/ACM Transactions on Networking*, vol. 12, no. 6, 2004.

[21] Mingzhe Li, Mark Claypool, and Robert Kinicki, "Packet Dispersion in IEEE 802.11 Wireless Networks," in *Proceedings Performance and Management of Wireless and Mobile Networks (P2MNet)*, Tampa, FL, USA, Nov. 2006.

[22] Feng Li, Mingzhe Li, Rui Lu, Huahui Wu, Mark Claypool, and Robert Kinicki, "Measuring Queue Capacities of IEEE 802.11 Wireless Access Points," in *Proceedings of the Fourth IEEE International Conference on Broadband Communications, Networks and Systems (BROADNETS)*, Raleigh, NC, USA, Sept. 2007.

[23] Y. Tobe, Y. Tamura, A. Molano, S. Ghosh, and H. Tokuda, "Achieving moderate fairness for udp flows by path-status classification," in *LCN '00*, Washington, DC, USA, 2000, p. 252, IEEE Computer Society.

[24] Saad Biaz and Nitin H. Vaidya, "Distinguishing congestion losses from wireless transmission losses : A negative result," *icccn*, vol. 00, pp. 722, 1998.

[25] Mingzhe Li, Mark Claypool, and Robert Kinicki, "WBest: a Bandwidth Estimation Tool for Multimedia Streaming Application over IEEE 802.11 Wireless Networks," Tech. Rep. WPI-CS-TR-06-14, Computer Science Department at Worcester Polytechnic Institute, July 2006.