WBest: a Bandwidth Estimation Tool for Multimedia Streaming Application over 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—
Multimedia streaming applications can benefit from bandwidth estimation techniques to perform media scaling and buffer optimization efficiently. However, most current techniques were designed for wired networks and produce relatively inaccurate results and long convergence times on wireless networks where capacity and contention for the capacity can vary dramatically. Therefore, it is difficult to apply current bandwidth estimation tools to multimedia streaming applications in wireless network. 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 applies a two-step algorithm: 1) a packet pair technique to estimate the effective capacity of the wireless networks; 2) a packet train technique to estimate the achievable throughput and report the inferred available bandwidth. Using an analytic model, the possible error sources are explored and WBest parameters are optimized given the tradeoffs of accuracy, intrusiveness and convergence time. The advantage of WBest is that it does not depend upon search algorithms to detect the available bandwidth but instead, statistically detects the available fraction of the effective capacity, mitigating estimation delay and the impact of random wireless channel errors. WBest is implemented and evaluated on an 802.11 wireless testbed. Comparing WBest with other popular 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

Multimedia streaming leads the list of Internet applications that are significantly impacted by the accuracy of bandwidth estimation techniques [1]. Both media scaling techniques [2], [3] and client side buffering [4], [5] rely on the bandwidth estimate of the underlying network. In response to congestion, media scaling aims to adjust the media transmission rate below the available bandwidth to minimize packet loss. Client side buffering sacrifices start-up delay to reduce the jitter effects and playback disruption caused by bandwidth oscillations along the flow path. By estimating the variance of the available bandwidth, the client side buffer can be dynamically adjusted to reduce the probability of buffer underflow.

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 wireless networks. Fluctuating wireless channel conditions cause variability in wireless capacity and available bandwidth, and other wireless factors such as reception signal strength and bit error rates (BER) due to path loss, fading, interference and contention limit the effective bandwidth that the wireless network actually provides. While providing satisfying results on wired networks, current bandwidth estimation tools have been shown [6], [7], [8], [9] to be adversely impacted by IEEE 802.11 wireless network conditions.

Tools that provide only capacity estimates are not useful for Internet applications that adjust their traffic rate in response to other concurrent flows. Moreover, applications such as multimedia streaming need an available bandwidth estimate within a few seconds to avoid client-side buffer underflows and to satisfy users waiting to use the application. This implies a much faster convergence time requirement than some bandwidth estimation tools provide. The variability of the wireless channel implies that multiple bandwidth estimation invocations are typically used within a single application stream. This adds an additional requirement that a bandwidth estimation tool must be minimally intrusive so as to not adversely impact the application's performance during measurements.

Therefore, the issues discussed above make it inefficient to apply current bandwidth estimation mechanisms to multimedia streaming applications in wireless networks. The multimedia streaming applications require some additional considerations when performing the bandwidth estimation in wireless networks, such as low intrusiveness, fast convergence time, and consistent convergence time under variable channel conditions. Appendix V-B discusses the requirements of wireless multimedia streaming applications on bandwidth estimation technique in details.

Most available bandwidth estimation techniques are designed 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)* [10], *pathload* [11] and *pathChirp* [12], 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*

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

(PGM) techniques, such as *Initial Gap Increase/Packet Transmission Rate (IGI/PTR)* [13] and *Spruce* [14], 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 has proposed improvements to bandwidth estimation specific to wireless networks. *ProbeGap* [8] uses the one-way delay gap to estimate the available bandwidth in broadband access networks including IEEE 802.11 networks. However, *ProbeGap* does not provide capacity estimation and needs to use third party capacity estimation tools. *DietTOPP* [15] uses a reduced TOPP algorithm with a modified search algorithm to determine available bandwidth in wireless networks. While improving the accuracy of bandwidth estimation in wireless networks, these techniques do not consider convergence time and intrusiveness.

Packet dispersion techniques, such as packet pair or packet train probing, are used to measure the end-to-end capacity of a network path. First introduced in [16], [17], [18], packet pair dispersion techniques have been enhanced via tools such as *bprobe/cprobe* [19], *sprobe* [20], *pathrate* [21], [22], and *CapProbe* [23]. Packet dispersion techniques send two or more packets back-to-back into the network. After the packets traverse the narrow link, the time dispersion between the two packets is linearly related to the narrow link capacity. Packet dispersion for capacity estimation is vulnerable to crossing traffic that interferes with probing packets and causes estimation errors. However, the amount of interference can be used to estimate the amount of crossing traffic.

The issues of inaccurate results, high intrusiveness and long convergence time make it difficult to apply current bandwidth estimation mechanisms to applications, such as multimedia streaming, over wireless networks and lead to the development of the Wireless Bandwidth Estimation tool (WBest). To address accuracy and convergence, WBest employs packet dispersion techniques to provide capacity and available bandwidth information for the underlying wireless networks. Our previous research [24] models packet dispersion behavior in wireless networks under varying conditions. Using an analytical model, two packet dispersion measures, *effective capacity* and *achievable throughput*, were shown to be suitable for wireless networks. Combining these two metrics, WBest employs a two-step algorithm to determine available bandwidth. In the first step, a packet pair technique estimates the effective capacity of the wireless network. In the second step a packet train scheme determines achievable throughput and infers available bandwidth. By modeling WBest, this paper investigates the tradeoffs of accuracy and convergence time, and possible sources of error to optimize the algorithm. Thorough evaluation in a wireless testbed shows WBest performs better in terms of accuracy, intrusiveness and convergence time than three current available bandwidth estimation tools: *IGI/PTR*, *pathChirp* and *pathload*. WBest fits the practical needs of many applications such as multimedia streaming that require low cost and accurate bandwidth estimations.

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. Finally, Section V provides conclusions and possible future work.

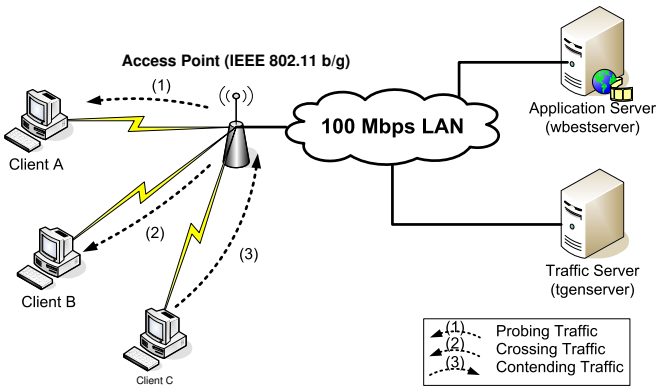


Fig. 1. Network Path with Last Mile Wireless Network.

II. WBest Algorithm

This section introduces WBest, an algorithm to estimate both effective capacity and available bandwidth on a network path where the last hop is over a wireless network.

Figure 1 depicts 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 mile' wireless connection. To provide better performance, such as to perform media scaling and buffer optimization for a multimedia stream, the application server needs to know the capacity and available bandwidth on the flow path. To characterize the wireless network impact for study, the network traffic is categorized as probing, crossing and contending, as depicted in Figure 1. *Probing traffic* is traffic sent by bandwidth estimation tools along the network path through the AP to the client (1). Wireless channel conditions and other traffic may affect the probing traffic behavior and produce estimation errors. *Crossing traffic* shares the bottleneck with the direction coming from the AP to associated clients (2). *Contending traffic* competes with probing traffic on the path being estimated when accessing the shared wireless channel. Contending traffic usually comes from clients to the same AP (3) or between other clients and APs within interference range, which is also known as co-channel interference due to neighboring APs. In addition to the bottleneck sharing effects, contending traffic causes further available bandwidth reduction due to wireless channel access contention. Capacity estimation should avoid estimation errors caused by crossing and contending traffic. However, available bandwidth estimation should capture the available bandwidth reduction due to both crossing and contending traffic.

A. Assumptions

To simplify the bandwidth estimation algorithm, the following assumptions are made. These assumptions and possible resultant errors are discussed in more detail later in this section.

1. Assume the last hop wireless network is the *bottleneck link* on the whole network path. Here the *bottleneck link* means the last hop wireless network has both the smallest available bandwidth (tight link) and the smallest capacity (narrow link) along the network path. That is we have the relationship:

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

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 at rate C_e is likely to arrive at the last hop at the rate of C_e [22]. If this assumption does not hold, as for some home wireless networks with a broadband Internet connection, the packet train with sending rate C_e will get dispersed before the last hop and arrive at the last hop with a lower rate than C_e. This, in turn, will cause a conservative under-estimate of the bandwidth which is typically a better outcome for applications than an aggressive, over-estimate.

2. Assume no significant changes in network conditions between the two steps (estimate effective capacity and estimate available bandwidth) of the WBest algorithm. While changes in network conditions due to rate adaptation or mobility may impact the estimation results, given algorithm convergence times of milliseconds, the magnitude of these changes is assumed to be minimal.

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 will impact the accuracy of the estimation results. The possibility of queuing loss is reduced by limiting the number of packet pairs and the number of packets in the packet train sent into the network.

B. Algorithm

Algorithm 1 provides the two-step WBest algorithm. In the first step (lines 1-2), n packet pairs are sent to estimate effective capacity C_e. Effective capacity [24], the maximum capability of the wireless network to deliver network layer traffic, is a function of time and the 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**

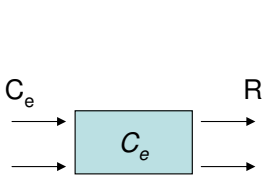


Fig. 2. A Typical Last Hop Wireless Network.

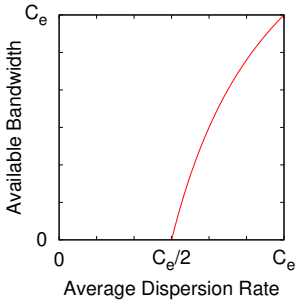


Fig. 3. Estimating Available Bandwidth using Average Dispersion Rate.

To minimize the impact of crossing and contending traffic, the median of the n packet pair capacity estimates is used to approximate C_e in the estimation time period:

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

where C_i is the estimation result of packet pair i and $C_i = \frac{L}{T_i}$.

For the second step of the algorithm (lines 3- 13), a packet train of length m is sent at rate C_e to estimate available bandwidth. A fluid model is used to estimate the relationship between available bandwidth and dispersion rate. Assumption 1 means the arriving rate before the last hop is C_e and:

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

where as shown in Figure 2, R is the average dispersion rate at the receiver and S represents the available bandwidth reduction caused by crossing and contending traffic at the last hop:

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

Combining Equations 4 and 5, the available bandwidth is:

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

For a wireless network, *achievable throughput* [24] is 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 packets losses occur to avoid errors in the estimation computation. However, this implies longer measurement times or at least more variance in 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 major advantages stem from statistically detecting the relative available fraction of the effective capacity in the wireless network instead of using search algorithms to measure

the absolute available bandwidth. Most current available bandwidth mechanisms detect absolute 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 determine clear packet delay trends. This reduces the accuracy and increases the convergence time, intrusiveness and instability of the estimation mechanism. 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 absolute rate, WBest estimates available bandwidth using the effective capacity. $(2 - \frac{C_e}{R})$ in Equation 6 is treated as the available fraction of C_e available to all wireless flows. Derived from the ratio of the effective capacity to the average dispersion rate, the available fraction statistically removes random errors while still capturing the impact of crossing/contending traffic and rate adaptations inherent in wireless networks.

C. Number of Packet Pairs and Length of Packet Train

The number of packet pairs in the first step of WBest and the number of packets in the packet train in the second step 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 modeled variance σ [24] can be used to estimate the minimum required number of packet pairs using:

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

where Z is a constant determined by the confidence level. For example, assume a streaming session wants to bound the effective capacity estimate within 500 Kbps to match the granularity of encoded scaling levels for the multimedia stream. To bound 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 rate and a packet size of 1500 bytes with $Z = 1.96$ and $CI = 500$ Kbps [24]. 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 [24] for an 11 Mbps channel link rate 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 provides an approximation on the sample sizes needed.

The number of packets in a train also impacts the time scale and sensitivity of available bandwidth estimations. In general, the estimation of available bandwidth represents the average estimation during the measurement period [1]. 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. 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 the 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 also limits the number of packet pairs and the length of the packet train. The *pathrate* queue size probing method [22] can be used to detect buffer limitations along the flow path. However, this probing method increases the 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 Access Point (AP), the probability of queue overflow in the network is determined by the queue size at the last hop wireless AP. Previous research [25] indicates that 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.

D. Impact of Errors in Effective Capacity Estimation

The effective capacity estimate in the first step of WBest impacts the available bandwidth estimate in the second step. If C_e' denotes the estimated effective capacity from the first step and C_e is the actual effective capacity, the fluid model from Equation 4 yields:

$$\frac{C_e'}{C_e' + C_e - A} = \frac{R}{C_e} \tag{9}$$

By defining the error ratio Y as $C_e' = C_e(1+Y)$, the dispersion rate is:

$$R = \frac{C_e' C_e}{C_e' + C_e - A} = \frac{(1+Y)C_e^2}{(2+Y)C_e - A} \tag{10}$$

Mimicking the derivation from Equation 6, the estimated available bandwidth, A', is $A' = 2C_e' - \frac{C_e'^2}{R}$. The relationship between estimated available bandwidth A' and real available bandwidth, A, then becomes:

$$A' = A(1+Y) - C_e Y(1+Y) \tag{11}$$

To study available bandwidth estimation errors due to error in effective capacity estimation, relative error, E, is defined as:

$$E = \frac{A' - A}{A} \tag{12}$$

Positive and negative values for relative error E denote overestimation and under-estimation of the available bandwidth, respectively. From Equation 12, the relative error in available

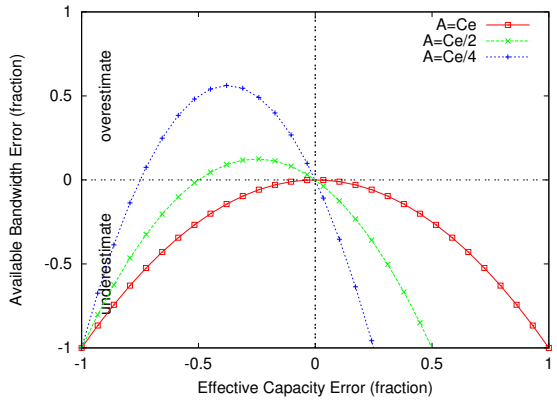


Fig. 4. Relative Error Caused by Effective Capacity Estimation Errors.

bandwidth estimation is $E = Y - Y(1+Y)C_e/A$. Figure 4 shows in fractional form the relative error of estimated available bandwidth and the relative error in capacity estimation for three distinct cases. Effective capacity estimations that are too high always under-estimate the available bandwidth. Effective capacity estimations that are too low can result in either an over-estimate or under-estimate of the available bandwidth, depending upon on the actual available bandwidth in the network. Errors in effective capacity estimation can be bounded by modeling [24] or by measurement, e.g., using the range of the results from the capacity estimation step to approximate the relative error on the estimation of available bandwidth. Moreover, for applications where a conservative estimate of the available bandwidth is desireable, such as in multimedia streaming, a higher effective capacity estimator can be used (e.g., using the top 10% in Equation 3 for C_e instead of the median) to minimize the potential performance degradation caused by over-estimating the available bandwidth of the underlying network.

D.1 Pre-dispersion and Pre-compression

We assumed that the last mile is the bottleneck of the network path defined in Equation 1. However, even though we can assume that the last hop have the less available bandwidth, we may still expect some problems such that the packet train arrive at the AP with a lower rate or higher rate than C_e. We call this rate R_p as pre-dispersion and pre-compression rate because they happen before the packet train arrive at the last hop wireless networks.

The possible sources of a pre-dispersion could be a link with an available bandwidth A_i less than the packet train rate C_e, which is also the effective capacity of the last hop wireless network. Therefore, to analyze the impact caused by the pre-dispersion and pre-compression behaviors, we can use the same fluid model as Equation 4 if the pre-dispersion/pre-compression rate R_p is greater than or equal to the available bandwidth A:

$$\frac{R_p}{R_p + C_e - A} = \frac{R}{C_e} \qquad (R_p \geq A) \tag{13}$$

In the case of pre-dispersion rate, R_p is less than the available bandwidth, the probing traffic will not be further dispersed at the wireless hop. Therefore, Equation 4 is not applicable and we have $R = R_p$. We define X as the ratio of pre-dispersion/pre-compression on the probing traffic, such that we have $R_p = C_e(1+X)$, where a positive X denotes a pre-compression and a negative X denotes a pre-dispersion. Following the same derivation we can get the dispersion rate after passing the last hop as:

$$R = \begin{cases} \frac{C_e R_p}{R_p + C_e - A} = \frac{C_e^2(1+X)}{(2+X)C_e - A} & (R_p \geq A) \\ C_e(1+X) & (R_p < A) \end{cases} \tag{14}$$

As described in Equation 6, the estimated available bandwidth with pre-dispersion or pre-compression is defined as $A' = 2C_e - \frac{C_e^2}{R}$. Thus, by representing R using C_e, A and X, we can derive the relation between estimated available bandwidth A' and real available bandwidth A from Equation 15 as:

$$A' = \begin{cases} \frac{A}{1+X} - \frac{XC_e}{1+X} & (R_p \geq A) \\ 2C_e - \frac{C_e}{1+X} & (R_p < A) \end{cases} \tag{15}$$

To study the errors caused by pre-dispersion and pre-compression, we compute the relative error E between the estimated available bandwidth and the real available bandwidth using Equation 12. Therefore, a positive and negative relative error E denote a over-estimation and under-estimation of the available bandwidth, respectively. The relative error in available bandwidth caused by pre-dispersion and pre-compression can be derived as Equation 16:

$$E = \begin{cases} \frac{X(C_e/A - 1)}{1+X} & (R_p \geq A) \\ \frac{(1+2X)C_e/A}{1+X} - 1 & (R_p < A) \end{cases} \tag{16}$$

Figure 5 shows theoretical relationship of pre-dispersion and pre-compression ratio X and the relative error in available bandwidth E for the network with different amount of available bandwidth.

It clearly shows that pre-dispersion results in a lower estimated available bandwidth than the real available bandwidth. On the contrary, pre-compression results in a higher estimated available bandwidth. Moreover, as the available bandwidth decreases, the impact caused by pre-dispersion and pre-compression increases. In addition to the theoretical relation, if the pre-dispersion reduces the packet train probing rate lower than the available bandwidth, there will be no further dispersion at the last hop. Therefore, Equation 13 cannot be used to compute the dispersion rate R. Instead, we have $R = R_p$ and the relative error of available bandwidth can be computed based on Equation 6 and 12. The "No dispersion" curve depicts the converting point of the relative errors when the pre-dispersion rate is lower than the available bandwidth.

For streaming applications, the underestimation of the available bandwidth caused by pre-dispersion make the media scaling more conservative, which is helpful for avoiding performance problems such as bursty lost and rebuffer events. The overestimation caused by pre-compression impacts the media scaling performance. A possible solution is to increase the estimation samples, thus reducing the errors caused by pre-compression. Be aware that the errors shown in Figure 5 represent the worst case such that all samples in the estimation are pre-dispersed/pre-compressed for the given ratio, we expect a lower relative error in practices because the WBest algorithm is based on an average of multiple samples.

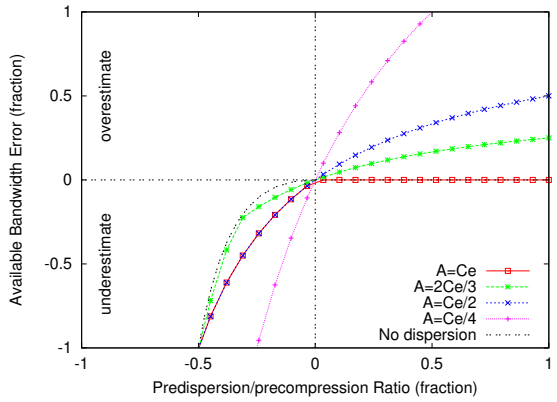


Fig. 5. Relative error caused by pre-dispersion/compression

E. Error Detection

Packet loss observed at the WBest 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 [26], [27] could be added to WBest to distinguish congestion loss or 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 [22] may cause two or more packets to have very close arrival timestamps. This 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, if 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 Linux with 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 Mbits 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 [12]. In this method, the difference between the 90th and 10th percentiles of the estimations 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 estimation accuracy, the true available bandwidth of the wireless network under different configurations is needed – referred to here as the *ground truth*. Since it is difficult to get the actual ground truth during dynamic wireless network conditions, 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. The exception is for cases with TCP crossing and contending traffic where ground truth for the available bandwidth in the wireless network is zero. Thus, each evaluation consists of back-to-back runs employing four bandwidth estimation tools and one downstream CBR traffic, as shown in Figure 6. 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 the wireless network conditions. Cases 1-12 include a variety of crossing and/or contending traffic situations provided by UDP and TCP traffic generators residing on client B, client C and tgenserver. The Multi-Generator Toolset (mgen) v4.2b6 [8] and iperf v2.0.2 [9] 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

[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 [12]
[8] http://pf.itd.nrl.navy.mil/mgen/
[9] http://dast.nlanr.net/Projects/Iperf/

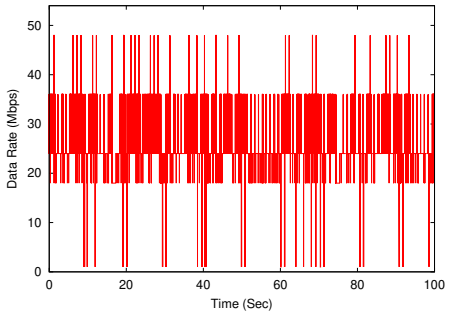


Fig. 7. Rate adaptation Behavior.

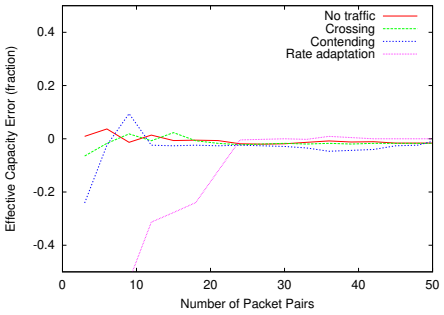


Fig. 8. Analysis of Number of Packet Pairs.

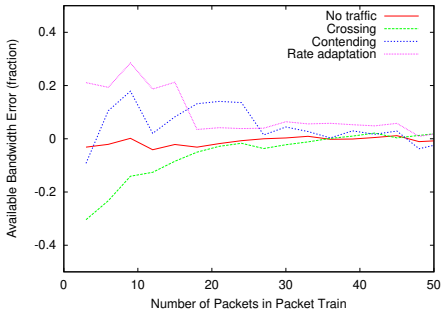


Fig. 9. Analysis of Length of Packet Train.

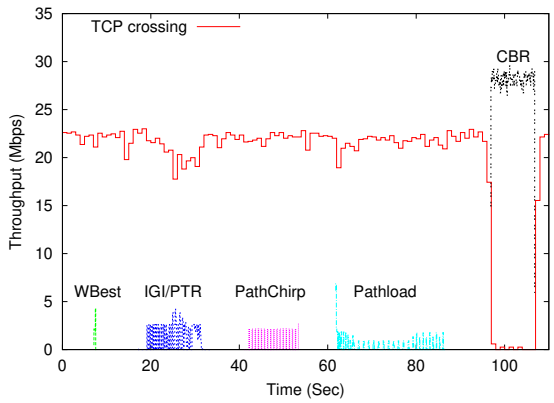


Fig. 6. An Example of a Sequential Run of Bandwidth Estimation Tools.

-70 dbm and -74 dbm, the rate adaptation ranged from 1 to 48 Mbps. Figure 7 shows the actual rate adaptation measured with a wireless sniffer[10]. This rate adaptation case results in 8% of 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 repeated 30 times with 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.

[10]http://perform.wpi.edu/tools/

To mitigate interference from co-existing campus wireless networks, all experiments are run in our wireless streaming multimedia lab[11] which was painted with an additive[12] to reduce the radio transmissions going through the walls. Furthermore, all the experiments were conducted at midnight during the WPI summer break such that most of the campus wireless network was in an idle state.

The relationship between relative error and the number of pair pairs in step 1 of the WBest algorithm (estimate effective capacity) was explored by using Equation 12 to compute the error of the estimated effective capacity using different numbers of packet pairs and defining real effective capacity as the median of the 90 packet pair run. Figure 8 shows the relationship between the effective capacity error and the number of packet pairs sent for four typical wireless cases: idle, crossing traffic, contending traffic, and rate adaptation. 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 these cases while reducing the impact on the available bandwidth estimations, 30 packet pairs were used in all the WBest evaluations. Similarly, based on Figure 9, 30 was chosen as the length of the packet train for step 2 of the WBest algorithm (estimate available bandwidth) for all the WBest experiments.

IV. ANALYSIS

A. Data Collected

For each of the fourteen test cases, Table II gives the median estimated available bandwidth for 30 evaluations runs of each of the four bandwidth estimation tools. The 'ground truth' column provides the true available bandwidth, approximated from the measured CBR UDP throughput with a packet size of 1500 bytes or set to zero if the specific test case includes a TCP bulk transfer as described in Section III.

For Case 6, the UDP traffic from the two contending clients causes the AP and the clients to use rate adaptation even with good RSSI values. While it is normal for rate adaptation to be triggered by high contention for the wireless channel, the saturated CBR throughput of 9.29 MBps for case 6 does not represent ground truth because higher throughput can be obtained with a lower offered CBR rate, as could be the case with the bandwidth estimation tools. Thus, for case 6 the ground truth is marked as unknown. Appendix V-D discusses the rate adapta-

[11]http://perform.wpi.edu/wsml/
[12]http://www.forcefieldwireless.com/defendairadditive.html

tion of case 6 in detail. In general, for all other cases in Table II, WBest provides the most accurate estimation of the available bandwidth compared to the other three bandwidth estimation techniques.

In addition to the accuracy, the intrusiveness and convergence time is recorded for each test case. The intrusiveness is defined as the total bytes sent by each tool during an estimation and the convergence time the time spent by each tool to converge to a bandwidth estimation result in each estimation. Table III provides the median of value of intrusiveness and convergence times over 30 runs for all fourteen test cases. WBest yields the lowest intrusiveness and convergence time in every case.

TABLE II
ESTIMATED AVAILABLE BANDWIDTH (MEDIAN, IN MBPS).

case	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	
case	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

B. Case analysis

Due to space limitations, detailed analysis is provided for only four representative cases from the set of fourteen experiments. Details are presented as box-and-whisker plots[13] for the idle channel (case 0), crossing traffic (case 1), contending traffic (case 2), and rate adaptation (case 13). The complete set of test results can be found in Appendix V-A.

C. Idle Channel (Case 0)

Figure 10 depicts the estimations, intrusiveness and convergence times for the idle channel (case 0). When the wireless channel is idle, the available bandwidth and the effective capacity are the same. The measured ground truth throughput shows the available bandwidth/effective capacity of 28.94 Mbps, close to the maximum throughput of 31.4 Mbps mentioned in Cisco document.[14] Figure 10 shows that *IGI/PTR* and *pathload* significantly under-estimate the available bandwidth. A possible reason is that the packet sizes used during probing 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 [24], [15], [8], so the maximum throughput will be lower for these smaller packet sizes. Since with a 500 byte or 200 byte packet, the maximum throughput of the wireless network is around 19.2 Mbps or 11.4 Mbps, respectively, even with the consideration of smaller packet sizes, *IGI/PTR* and *pathload* still significantly underestimate the 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 variance in the estimation. *Pathload* and *pathChirp* both have long convergence times, because both apply a search algorithm to adapt the probing rate during the estimations.

D. UDP crossing traffic (Case 1)

Figure 11 depict the estimations, intrusiveness and convergence times when there is one UDP crossing traffic flow (case 1). WBest performs better than the other tools with low intrusiveness and convergence times and accurate estimated 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.

E. UDP contending traffic (Case 2)

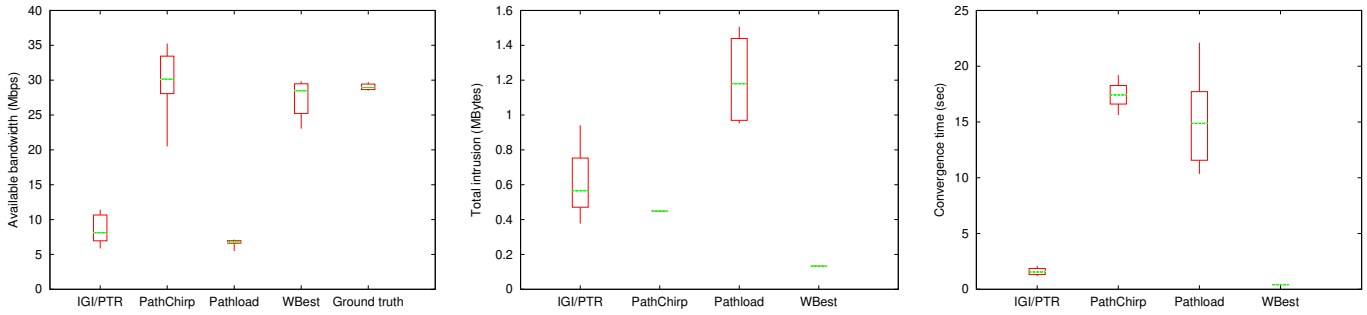
Figure 12 shows results when there is one UDP contending flow (case 2). WBest still performs well in the presence of contending traffic, however the variance is larger than in the case of the crossing traffic (case 1), because contending traffic increases the variance in delay in accessing the wireless channel. Since the sampling period is smaller in WBest than in other tools, the variance is amortized by other tools, such as *pathload*. Again, comparing case 2 with with case 0 and 1, *IGI/PTR* and *pathload* are not sensitive to contending traffic.

F. Wireless rate adaptation (Case 13)

Figure 13 shows results for wireless rate adaptation (case 13), where the the packet transmision rate and channel access delay vary as in Figure 7. With wireless rate adaptation, all the bandwidth estimation tools produce a larger variance than when there

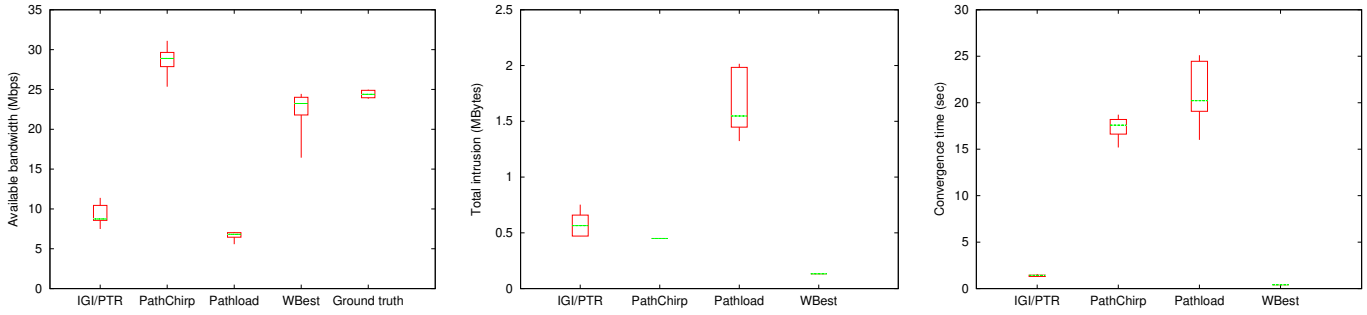
[13] In a box-and-whisker plot, the ends of the box are the upper and lower quartiles, the horizontal line inside the box is the median and the two lines (whiskers) outside the box extend to the 10 and 90%-tile of the observations.

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



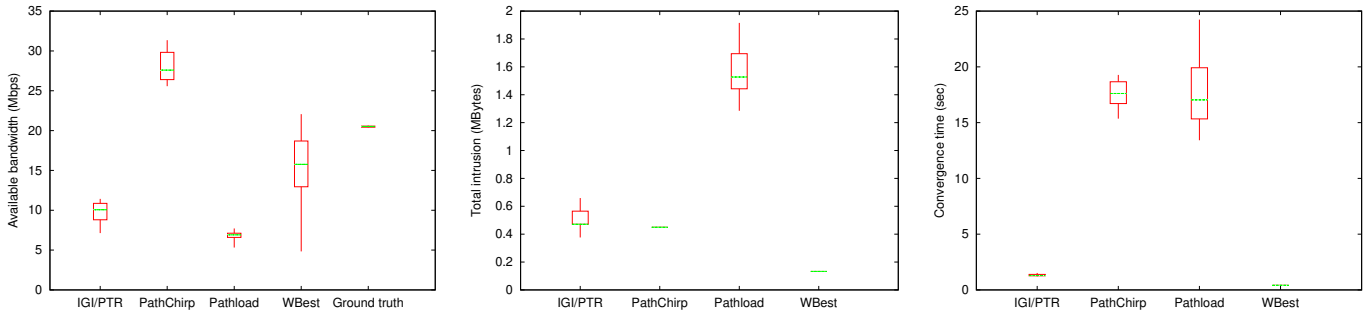
(a)Estimated available bandwidth (Mbps) (b)Intrusiveness (MBytes) (c)Convergence Time (Seconds)

Fig. 10. Summary Results for Evaluation Case 0 (Idle Channel).



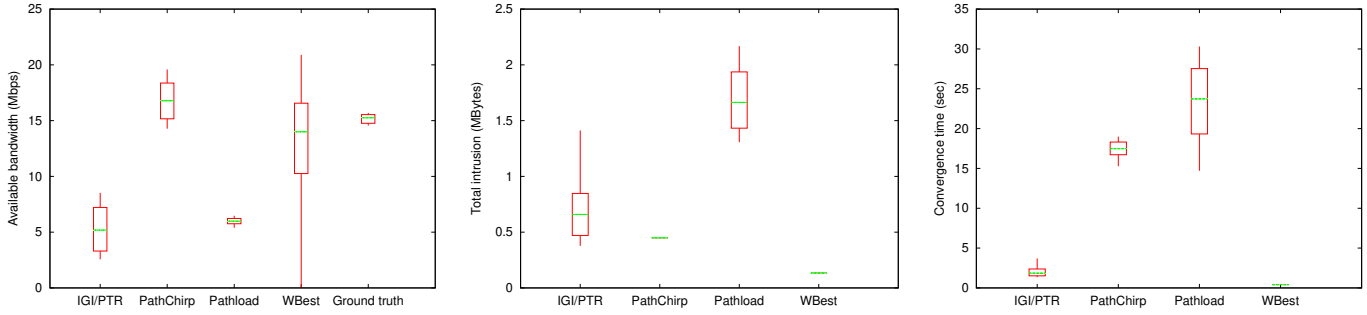
(a)Estimated available bandwidth (Mbps) (b)Intrusiveness (MBytes) (c)Convergence Time (Seconds)

Fig. 11. Summary Results for Evaluation Case 1 (One UDP Crossing Traffic of 4.6 Mbps).



(a)Estimated available bandwidth (Mbps) (b)Intrusiveness (MBytes) (c)Convergence Time (Seconds)

Fig. 12. Summary Results for Evaluation Case 2 (One UDP Contending Traffic of 4.6 Mbps).



(a)Estimated Available Bandwidth (Mbps) (b)Intrusiveness (MBytes) (c)Convergence Time (Seconds)

Fig. 13. Summary Results for Evaluation Case 13 (Wireless Rate Adaptation, Range 1-48 Mbps).

is no rate adaptation. However, *pathload*'s variance remains low with rate adaptation.

G. Summary

To provide summary analysis, the estimation error of each case is computed and the distributions of the error versus the convergence time and error versus intrusiveness are drawn in Figure 14 and 15, respectively. For 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. Therefore, good, fast estimates lie in the bottom center of these two figures.

IGI/PTR tends to greatly under-estimate the available bandwidth with UDP crossing or contending traffic and even with an

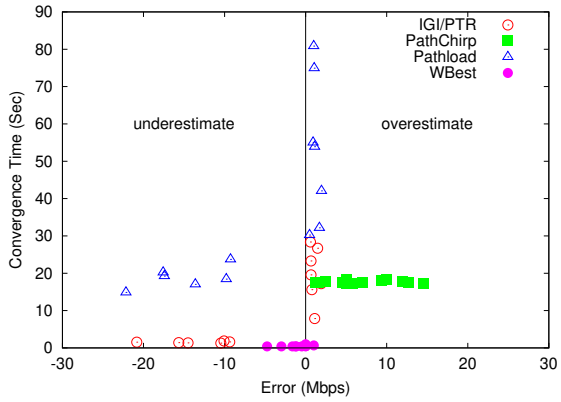


Fig. 14. Summary of All Experiments – Convergence Time versus Error.

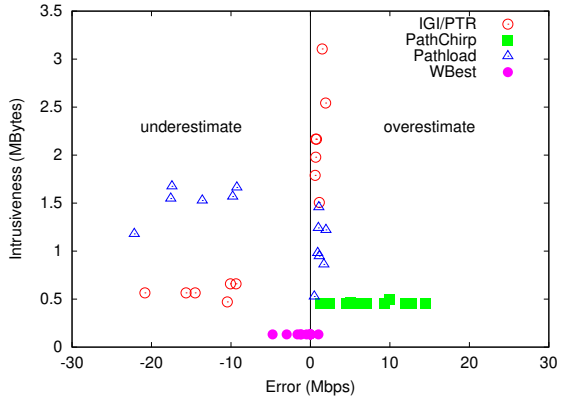


Fig. 15. Summary of All Experiments – Instrusiveness versus Error.

idle channel. *IGI/PTR* has widely variable convergence times and intrusiveness, varying by a factor of 20 times for the different cases. *PathChirp* tends to over-estimate the available bandwidth in all cases. *PathChirp* has a consistent convergence time of around 17 seconds and a consistent intrusiveness of about 400 KBytes. *Pathload* tends to greatly under-estimate the available bandwidth in most wireless traffic 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 even fails to converge in 100 seconds for some crossing and contending cases. WBest generally provides the most accurate estimations compared with the other tools. In most cases, WBest converges in less than half a second, and has a nearly constant instrusiveness of 130 KBytes.

For wireless networks, the accuracy of *IGI/PTR*, *pathChirp* and *pathload* is poor because each approach relies on delay changes to measure available bandwidth. In wireless networks queuing delay is not the only source of changes in delay. Wireless contention, MAC layer retries and rate adaptation can all result in delay changes to different extents. These delay changes disturb the searching algorithm for these tools and yield inaccurate results and often increase the convergence times and intrusiveness. Moreover, with higher packet loss rates in wireless networks, some estimation techniques discard probes impacted by loss to improve accuracy, but this also increases convergence time and intrusiveness.

WBest estimates the available bandwidth without using searching algorithms which means a low, consistent convergence time and intrusiveness. Furthermore, WBest does not depend on delay measurements to detect the available bandwidth. Instead, WBest detects the available bandwidth in terms of fraction of the effective capacity by measuring the relative changes in packet dispersion between two steps. This makes WBest robust even when packet dispersion is impacted by the wireless conditions.

V. Conclusion

This paper presents WBest, a new bandwidth estimation tool for wireless networks, designed to provide accurate bandwidth estimation in a short amount of time and without excessively intruding on existing traffic. One advantage of WBest over existing tools is that WBest does not depend upon search algorithms to measure available bandwidth. Instead, WBest statistically measures the relative available fraction of the effective capacity, mitigating estimation delay and the impact of wireless channel errors. WBest is compared with other popular available bandwidth estimation tools in a wireless testbed under a variety of wireless and network conditions. The following conclusions can be drawn:

1. Current bandwidth estimation tools are signficantly impacted by wireless network conditions, such as contention from other traffic and rate adaptation. This results in inaccurate estimates and high and varying convergence times and intrusiveness. This makes current tools generally impractical for applications running over a wireless link, such as streaming media, that require fast, accurate, non-instrusive bandwidth estimates.
2. WBest consistently provides fast available bandwidth estimation, with overall more accurate estimations and lower instrusiveness over all conditions evaluated.

Our ongoing work is to apply WBest to multimedia streaming applications to improve the performance of media scaling and buffer optimization in wireless networks. Other possible future work may include 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 assumptions inherent in Figure 4 and to enhance WBest robustness during AP queue overflow.

References

[1] R.S. Prasad, M. Murray, C. Dovrolis, and K.C. Claffy, "Bandwidth Estimation: Metrics, Measurement Techniques, and Tools," *IEEE Network*, November-December 2003.

[2] Paul Bocheck, Andrew Campbell, Shih-Fu Chang, and Raymond Lio, "Utility-based Network Adaptation for MPEG-4 Systems," in *Proceedings of NOSSDAV*, Basking Ridge, NJ, USA, June 1999.

[3] Avanish Tripathi and Mark Claypool, "Improving Multimedia Streaming with Content-Aware Video Scaling," in *Workshop on Intelligent Multimedia Computing and Networking (IMMCN)*, Durham, North Carolina, USA, Mar. 2002.

[4] Bill Birney, "Reducing Start-up Latency with Windows Media 9 Series," Feb. 2004, Microsoft Knowledge Base.

[5] Kang Li, Charles Krasic, Jonathan Walpole, Molly H. Shor, and Calton Pu, "The minimal buffering requirements of congestion controlled interactive multimedia applications," in *IDMS '01: Proceedings of the 8th International Workshop on Interactive Distributed Multimedia Systems*. 2001, pp. 181–192, Springer-Verlag.

[6] Samarth Shah, "Available bandwidth estimation in IEEE 802.11-based wireless networks," in *Finale Report of Bandwidth Estimation ISMA Workshop*, San Diego, CA, USA, Dec. 2003.

[7] 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.

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

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

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

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

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

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

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

[19] 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.

[20] 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/.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] Mark Claypool, Robert Kinicki, Feng Li, Mingzhe Li, Rui Lu, and Huahui Wu, "Measuring the queue sizes of ieee 802.11 wireless access points," Tech. Rep., CS Department at Worcester Polytechnic Institute, May 2006.

[26] 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.

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

[28] M. Zink, O. Kunzel, J. Schmitt, and R. Steinmetz, "Subjective Impression of Variations in Layer Encoded Videos," in *Proceedings of the 11th International Workshop on Quality of Service (IWQoS)*, Monterey, CA, USA, June 2003.

[29] Mingzhe Li, Feng Li, Mark Claypool, and Robert Kinicki, "Weather Forecasting - Predicting Performance for Streaming Video over Wireless LANs," in *International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Skamania, Washington, USA, June 2005.

[30] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Throughput: A Simple Model and Its Empirical Validation," in *Proceedings of ACM SIGCOMM*, Vancouver, Brisish Columbia, Candada, 1998.

[31] Amit P. Jardosh, Krishna N. Ramachandran, Kevin C. Almeroth, and Elizabeth M. Belding-Royer, "Understanding Congestion in IEEE 802.11b Wireless Networks," in *Proceedings of IMC*, Berkeley, CA, 2005.

APPENDIX

A. Extra results

This section shows the extra experiment results that were not included inline in the paper (for space constraints for a conference submission). Figure 16 to Figure 25 shows the box-whisker figures of estimated available bandwidth, intrusiveness and convergence time for experiment case 3 to 12, respectively.

B. Bandwidth Estimation for Streaming Applications in Wireless Networks

Bandwidth estimation techniques have been widely studied in recent years. However, there are few studies discuss issues of applying such techniques in real applications. Different applications and network environments may have distinct requirements on bandwidth estimation, thus need diverse adaptations in bandwidth estimation tools. It is difficult to design a general purpose bandwidth estimation tool for all type of applications. Therefore, to evaluate the applicability of a bandwidth tool we should include the applications and the network context.

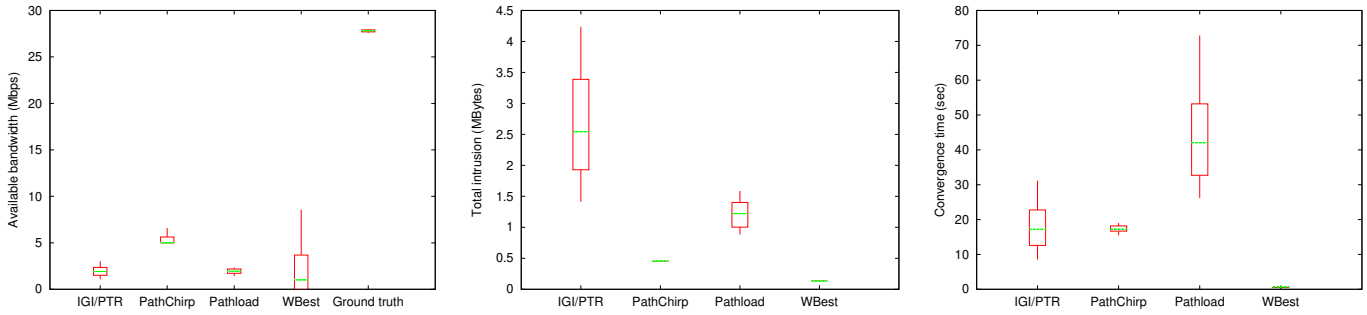
The previous bandwidth estimation tools usually target on network management, monitoring system, thus prefer to have accuracy bandwidth results. The applied bandwidth metrics may be either capacity or available bandwidth of the backbone networks. To describe the differences between the bandwidth estimation tool required by multimedia streaming applications and the general purpose bandwidth estimation tools, we follows the generate applied evaluation criteria: measured metrics, accuracy, convergence time, intrusiveness, robustness and usability in the wireless networks.

The bandwidth metrics used in bandwidth estimation tools in wired network need to be redefined in wireless network. For example, the capacity is not constant in wireless networks [24], instead, the *effective capacity* [24] that takes the dynamical capacity changes into consideration is more preferred. Similar, as discussed in Section II, the available bandwidth is not defined as the capacity excludes the amount cross traffic, instead, it should take the consideration of both capacity sharing and contending effects. However, most current available bandwidth estimation tools did not study these issues, for example, IGI [13], assumption of a known constant capacity. This not true for wireless networks, thus need to be adapted before applying to wireless networks. For a multimedia streaming application using bandwidth estimation to adapt the sending rate and optimize the client side buffer, the bandwidth metrics that helps are the available bandwidth and the statistical information of available bandwidth, such as variance. Therefore, the capacity estimation only tools are not qualified to be used by streaming applications. Moreover, these tools designed only for wired networks need to be improved before they can be applied to streaming applications in wireless networks.

For multimedia streaming applications, the accuracy is not a primary concern any longer. The reason is that the streaming media applications usually scales the sending rate in steps instead of smoothly. Thus, any bandwidth estimation result with a granularity less than the streaming media encoding level steps is sufficient for controlling the media scaling. For example, for a media stream encoded at multiple levels of 700 Kbps, 1.2 Mbps, 2.5 Mbps and 5 Mbps, an available bandwidth estimation of 3.54 Mbps will trigger a media scaling down to 2.5 Mbps. However, any estimation result between 2.5 Mbps and 5 Mbps will trigger the same media scaling. Thus a maximum acceptable estimation error for the 3.54 Mbps estimation result can be computed as $min(3.54 - 2.5, 5 - 3.54) = 1.04$ Mbps, which indicates you do not need a accuracy lower than 1.04 Mbps in this case. In addition, a higher media scaling frequency implies a lower perceived quality [28]. Since both the effective capacity and the available bandwidth change dynamically in wireless networks, to reduce unnecessary media scaling actions, a time-based average measurement is preferred more than an accurate instantaneous bandwidth measurement. Therefore, the instantaneous accurate estimate is not a critical requirement for streaming media applications.

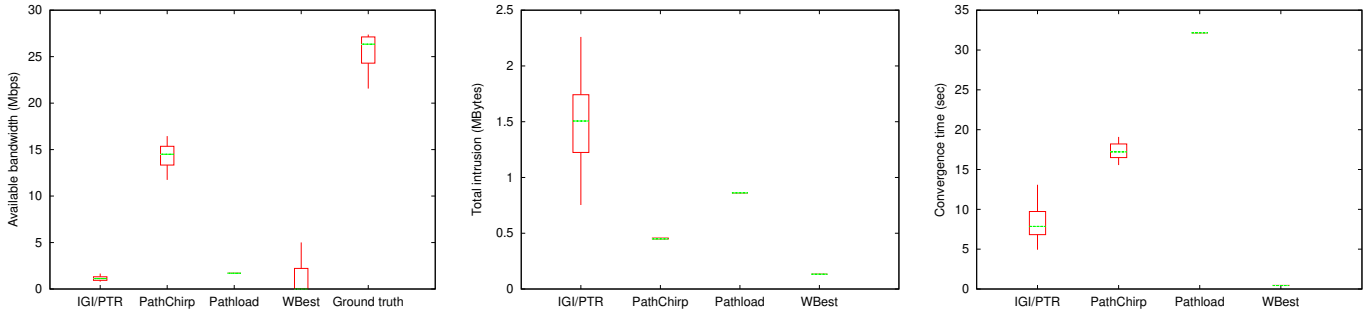
The convergence time is of major concern for streaming media applications over wireless networks. The application expects to know the available bandwidth change as soon as possible, even if the streaming media scaling does not need to execute at the same frequency. In addition, a short convergence time may provide more estimations in the same time period, thus may provide a better chance for a filtering or smoothing algorithm to find a reasonably accurate average estimation of bandwidth. A shorter convergence time improves the capability to capture the variation in the effective capacity or available bandwidth. As shown in recent research [29], the variation in the wireless link capacity may degrade the video performance even if the average capacity is sufficient for the streaming bit rate.

Another important issue related to both accuracy and convergence time is the *competing effects* caused by the probing traffic. With self-loading or packet dispersion techniques, the probing traffic will temporary increase the queuing delay of the crossing traffic. The responsive crossing traffic, such as TCP flows, will response to this RTT changes to reduce the sending rate. Therefore, the finally available bandwidth estimation will be overestimated if the convergence time is longer enough for TCP flow to reduce the sending rate. As discussed in [30], TCP throughput B can be approximate as the equation: $B = 1/RTT\sqrt{3/2bp}$, where p is the probability that a packet is lost, b is the number of packets that are acknowledged by a received ACK. Therefore, if the RTT is increased because of the probing traffic, the throughput of TCP traffics that sharing the same AP will be impacted. If we assume that the probing traffic will not overflow the queue, the TCP congestion control will response to the RTT changes to reduce it congestion window, thus will reduce the TCP throughput in few RTTs. For example, the time for a packet train with 30 packets to pass an AP with effective capacity of 6 Mbps is about 58 ms. If we assume the TCP RTT is in the same range, the throughput of the TCP traffic will decreased to almost half according to the equation. Therefore, we expected that the bandwidth estimation can be completed in less than few RTT so that the TCP crossing traffic will not back off due to the temporarily congestion caused by probing traffic. In fact, this competing effect is not particular for streaming applications or wireless networks, but for all bandwidth estimation tools. Possible solutions include reduce the intrusiveness and convergence time, or to approximate the amount of overestimation and compensate it to the final estimation of available bandwidth.



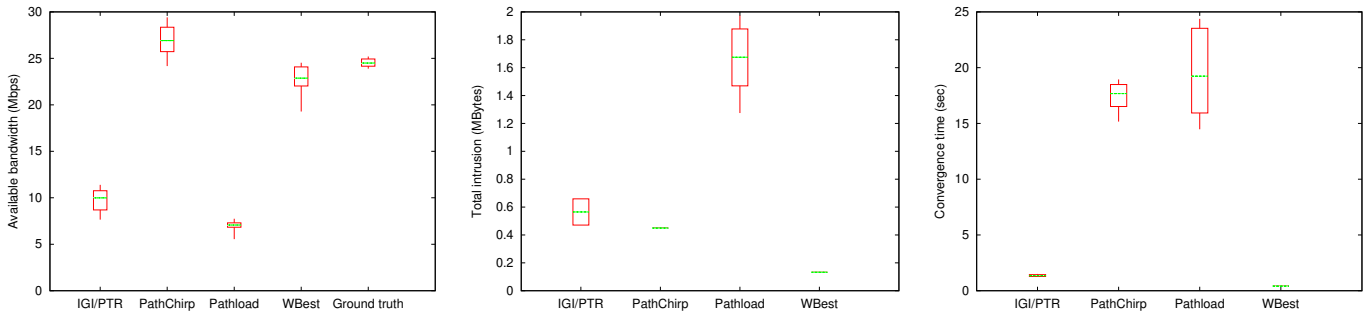
(a)Estimated available bandwidth (Mbps) (b)Intrusiveness (MBytes) (c)Convergence Time (sec)

Fig. 16. Summary Results of Evaluation Case 3 (One TCP Crossing Traffic)



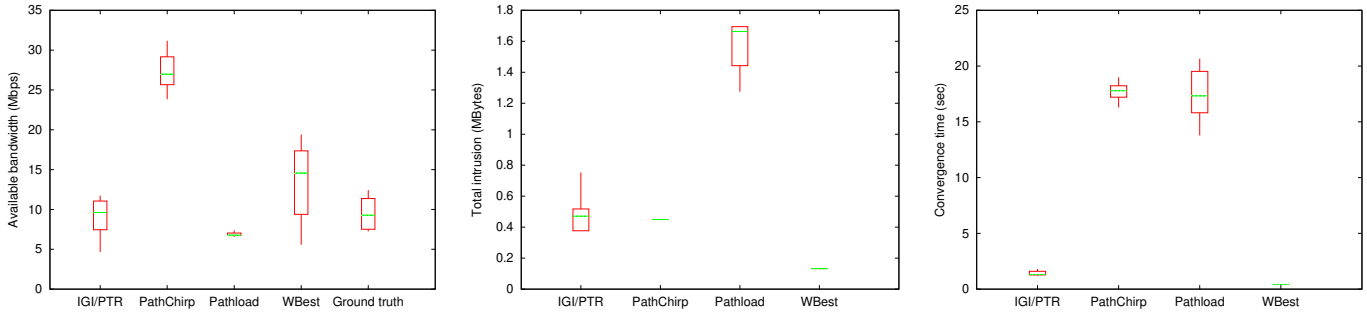
(a)Estimated available bandwidth (Mbps) (b)Intrusiveness (MBytes) (c)Convergence Time (sec)

Fig. 17. Summary results of Evaluation Case 4 (One TCP Contending Traffic)



(a)Estimated available bandwidth (Mbps) (b)Intrusiveness (MBytes) (c)Convergence Time (sec)

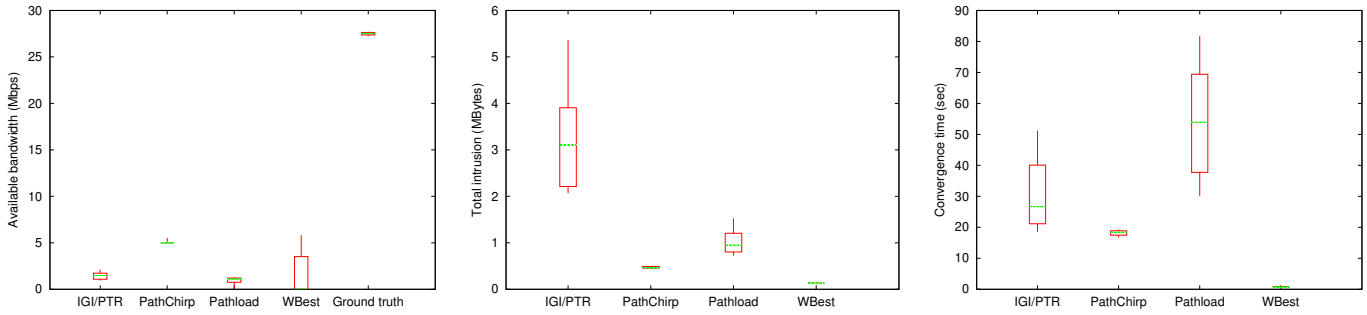
Fig. 18. Summary results of Evaluation Case 5 (Two UDP Crossing Traffic of 2.3 Mbps each)



(a)Estimated available bandwidth (Mbps) (b)Intrusiveness (MBytes) (c)Convergence Time (sec)

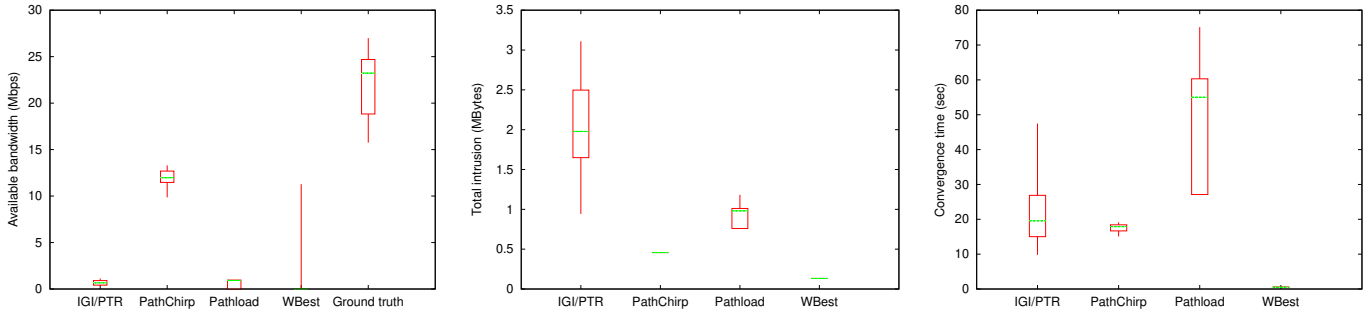
Fig. 19. Summary results of Evaluation Case 6 (Two UDP Contending Traffic of 2.3 Mbps each)

Intrusiveness is another major concern for evaluating the bandwidth estimation techniques over wireless networks. Streaming applications tend to perform bandwidth estimation frequently during the whole streaming session, therefore, a



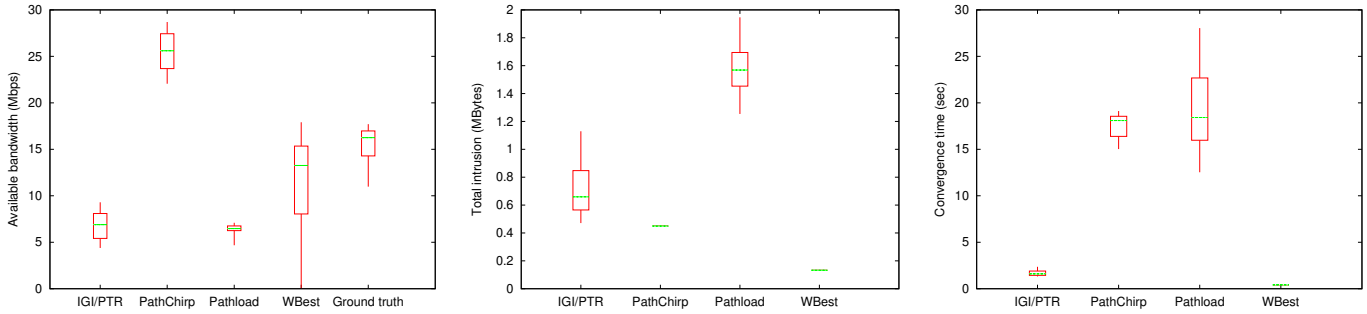
(a)Estimated available bandwidth (Mbps) (b)Intrusiveness (MBytes) (c)Convergence Time (sec)

Fig. 20. Summary results of Evaluation Case 7 (Two TCP Crossing Traffic)



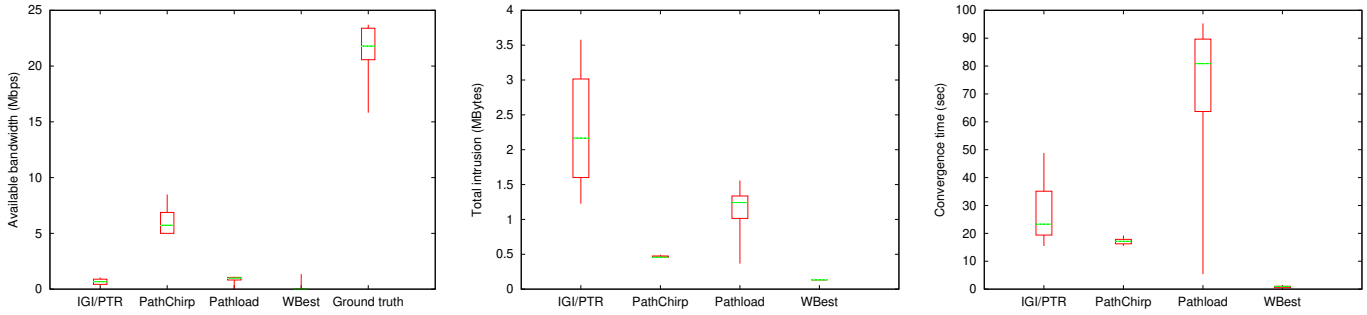
(a)Estimated available bandwidth (Mbps) (b)Intrusiveness (MBytes) (c)Convergence Time (sec)

Fig. 21. Summary results of Evaluation Case 8 (Two TCP Contending Traffic)



(a)Estimated available bandwidth (Mbps) (b)Intrusiveness (MBytes) (c)Convergence Time (sec)

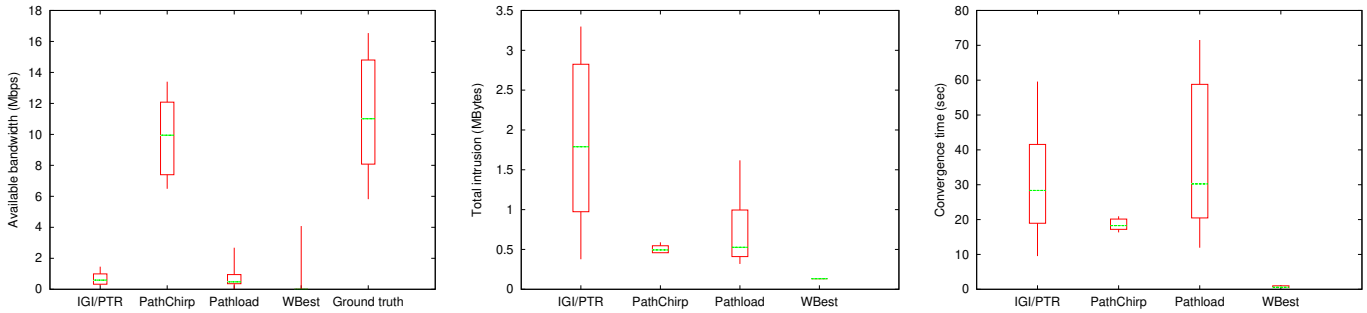
Fig. 22. Summary results of Evaluation Case 9 (One UDP Crossing and One UDP Contending Traffic of 2.3 Mbps each)



(a)Estimated available bandwidth (Mbps) (b)Intrusiveness (MBytes) (c)Convergence Time (sec)

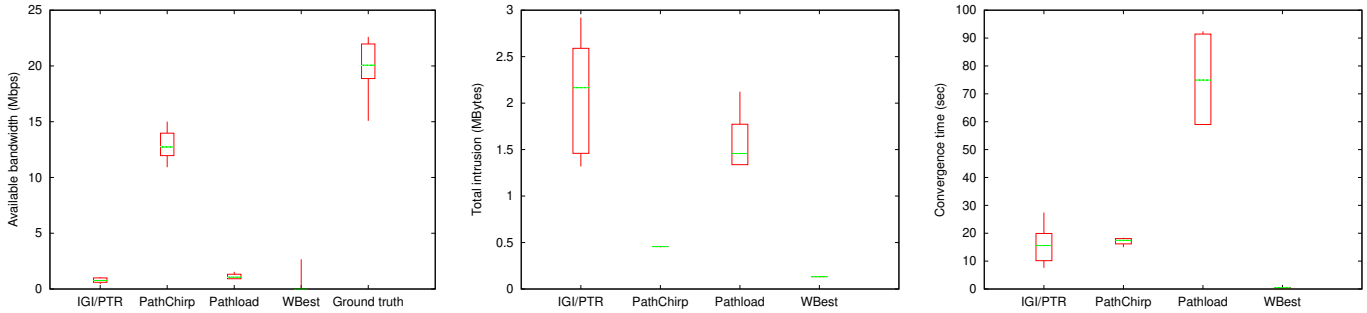
Fig. 23. Summary results of Evaluation Case 10 (One TCP Crossing and One TCP Contending Traffic)

lower intrusiveness is critical for reducing the impact caused by the probing traffic itself. Available bandwidth of wireless networks may be reduced due to the probing traffic over saturating the wireless network. As a result of bandwidth reduction, the



(a)Estimated available bandwidth (Mbps) (b)Intrusiveness (MBytes) (c)Convergence Time (sec)

Fig. 24. Summary results of Evaluation Case 11 (One UDP Crossing of 2.3 Mbps and One TCP Contending Traffic)



(a)Estimated available bandwidth (Mbps) (b)Intrusiveness (MBytes) (c)Convergence Time (sec)

Fig. 25. Summary results of Evaluation Case 12 (One TCP Crossing and One UDP Contending Traffic of 2.3 Mbps)

performance of streaming media applications can also be impacted by the heavy probing traffic.

Robustness and usability are mandatory to all bandwidth estimation techniques. Since streaming servers and clients are designed to work in client/server mode, the usability of working in uncooperative environments is not an issue any longer. However, to assure the applicability to streaming applications, the bandwidth estimation tool should have a relative consistent convergence time under varies channel conditions. That is, the applications expect the convergence time and intrusiveness to be bounded by upper limits, so that the applications can expected cost in term of time and intrusiveness on performing bandwidth estimation.

In summary, the multimedia streaming application in wireless network require a bandwidth tool with fast convergence time, low intrusiveness, reasonable accuracy, and consistent cost. For the initializing bandwidth estimation of multimedia applications, we expected the bandwidth estimation can be complete in few RTT time so that it will not add extra delay to the starting delay of the streaming. For the estimation during the streaming, we expected convergence times smaller than the buffer time of the streaming application, which means a in time adaptation before the buffer underflow. Given the expected convergence time of few RTT, most of the general purpose bandwidth estimation are not qualify for multimedia applications.

C. Implementation Issues

We implemented WBest on Linux system and evaluated it in our IEEE 802.11 wireless testbed. Even though Section II discusses issues that may impact the performance of WBest, there are additional issues in the implementation phase that may affect WBest as well.

The Linux system provides timers with millisecond (*sleep*) and microsecond (*usleep* and *select*) resolution timers. However, these timers may not satisfy the required resolution to control accurate sending rates. Therefore, we implement a busy-waiting timer using the *getimeofday* to provide microsecond resolution timer for the high sending rate cases. Even though the busy-waiting method may increase the CPU usage of the server during measurement, the impact caused by this short measurement duration is not significant, especially when the sending hosts of WBest are usually on high performance, multiprocessor servers. The microsecond resolution timer together with the *select* functions provide a reasonable sending rate control for WBest. Figure 26 shows an evaluation of the sending rate control mechanism of WBest. The mean sending rate for and the confidence interval shows that the rate control mechanism works as expected. Also, the CPU usage does not have noticeable increases when WBest is sending at the rate of 35 Mbps on a Pentinum 4 2.8 GHz computer, where 35 Mbps is about the maximum effective throughput of IEEE 802.11g working at 54 Mbps link data rate with a packet size 1460 Bytes.

D. Discussion on Experiments Setup

This section provide additional information about evaluation case 6, 13 and 14. As discussed in Section IV, case 6 with 2 UDP contending traffic flows experiences the impact of rate adaptation. Depending on the implementation of the rate adap-

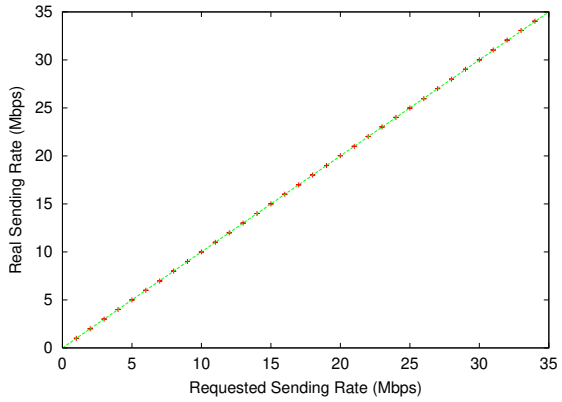


Fig. 26. Evaluation of sending rate control mechanism in WBest

tation algorithm, it is a normal behavior that wireless connections reduce the data rate upon multiple transmission failures, which could be caused by either low signal strength, high BER or contention. The rate adaptation triggered by contending effects may reduce the wireless network performance, which is also demonstrated in research [31]. To describe the rate adaptation behaviors of the clients and AP, we collect and analyze data from the wireless AP log and the packet captured on clients. Figure 27 and Figure 28 show that both the clients and AP the involve rate adaptation in a typical run of case 6. The AP log also denotes that AP has a high retry rate more than 20% with multiple retry data rate as shown in Figure 29. The throughput measured at one of the clients also confirm the impact caused rate adaptation as shown in Figure 30.

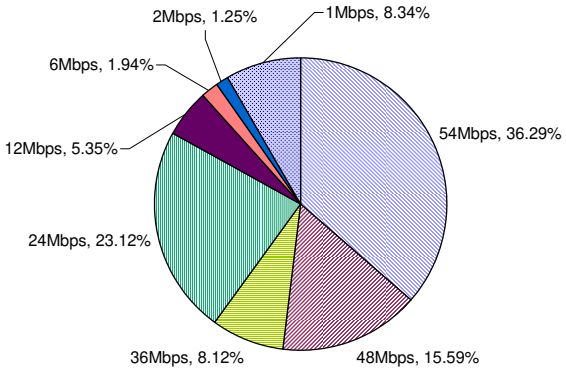


Fig. 27. Clients' data rate with two contending UDP traffic

Evaluation case 13 is designed to test WBest under rate adaptation conditions. As discussed in Section III, the rate adaptation is observed by a wireless sniffer. However, to show the impact on the packet delay, which could impact accuracy of delay-based bandwidth estimation, we show the RTT measured by ping with 64 byte and 1460 byte packet in Figure 31. Fewere than 10% of packets have a large RTT, where for the bad condition, in which the data rate is adapted to the channel condition, more than 40% of the packets have a large RTT. The RTT changes under the rate adaptation condition could potentially impact the accuracy of the delay-based bandwidth estimation tools.

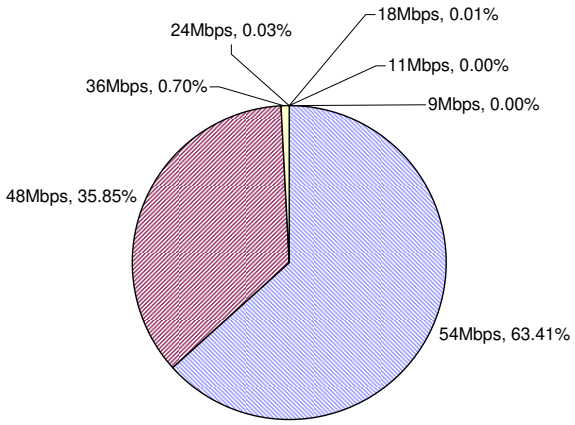


Fig. 28. AP's data rate with two contending UDP traffic

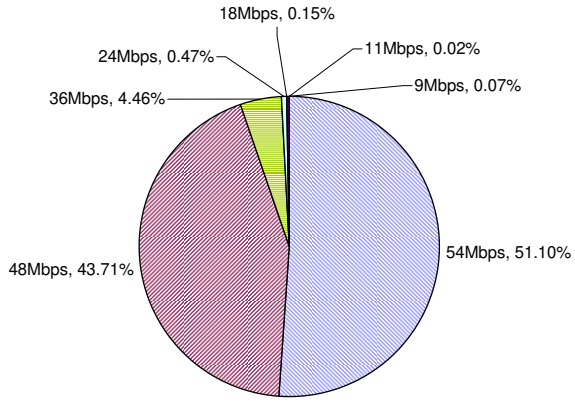


Fig. 29. AP retries' data rate with two contending UDP traffic

E. Extra Error Analysis

This section provides extra error analysis for WBest tools. The errors are computed using Equation 12. For these cases with 0 available bandwidth, the error is computed for estimated crossing/contending effects. Figure 32 and 33 depicts the cumulative distribution of the error in effective capacity and available bandwidth estimation for all cases evaluated. Both the effective capacity and available bandwidth have consistent estimations according to the CDF shown in the figures. Figure 34 shows that the relationship between effective capacity error and available bandwidth error. Even though the figure confirm that the underestimation in effective capacity could result in either over- or under-estimation in available bandwidth, it does not confirm that the overestimation in effective capacity always results in underestimation in available bandwidth. This is because the ground truth used in the calculation is the median of multiple runs, which could vary for each individual test, thus could vary the error computation.

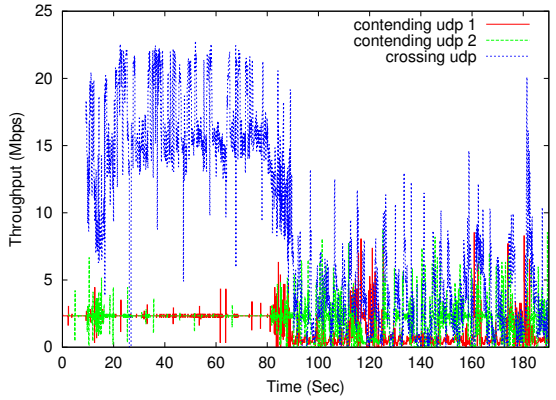


Fig. 30. Downstream UDP throughput with two UDP contending traffic

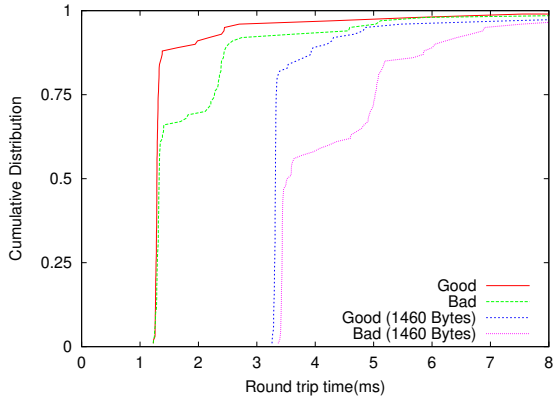


Fig. 31. Ping result of rate adaptation case

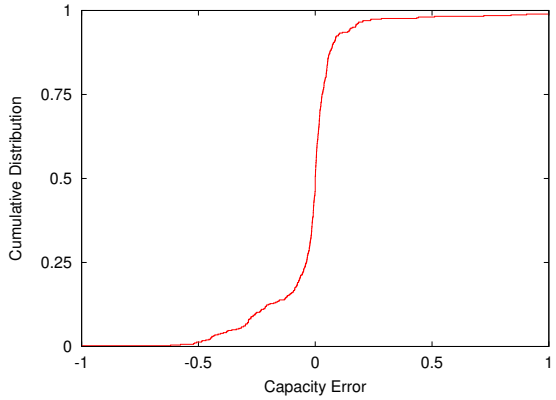


Fig. 32. CDF of capacity error

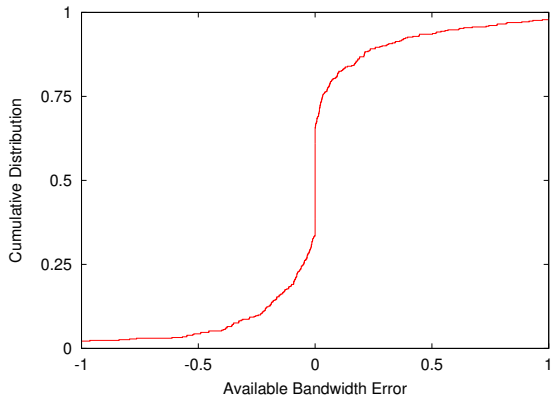


Fig. 33. CDF of available bandwidth error

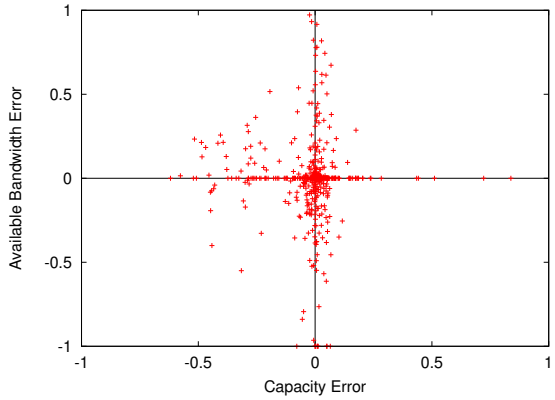


Fig. 34. Capacity error vs. available bandwidth error