RT-WABest: A novel end-to-end bandwidth estimation tool in IEEE 802.11 wireless network

Tan Yang¹,², Yuehui Jin¹, Yufei Chen¹ and Yudong Jin¹

Abstract
In 802.11 wireless network, a large number of multimedia applications are emerging. Accurate end-to-end bandwidth estimation techniques can help these applications adjust their behavior accordingly, such as stream’s bitrate adaptation, to improve quality of service. However, most current bandwidth estimation techniques are designed for wired networks that have high intrusiveness and consume long convergence time, which cannot be applied in resource-limited wireless network. In this article, we present a novel bandwidth estimation tool for IEEE 802.11 wireless network, which only needs to be deployed in one end of the end-to-end path and estimate available bandwidth based on round-trip-time measurements with low intrusiveness and short convergence time. We compare our tool with another state-of-the-art tool on an IEEE 802.11 wireless testbed, and the experimental results show that our tool has higher accuracy.

Keywords
Wireless network, packet dispersion, path capacity, available bandwidth, round-trip-time measurements

Date received: 18 September 2016; accepted: 30 January 2017

Academic Editor: Jose Moreno

Introduction
As many network applications and services have certain requirements of bandwidth, available bandwidth estimation is of great importance. For example, multimedia streaming services require available bandwidth estimation to adjust stream’s bitrate or sending rate dynamically and to improve quality of service (QoS). The available bandwidth estimation tools should adapt to the real-time changes of network with rapid convergence and reduce bandwidth consumption with minimal intrusiveness.

A number of bandwidth estimation tools have been developed in recent years, and they can be classified into two categories, one being passive estimation and the other being active probing. Passive estimation is done on the basis of congestion situation, packet loss, and delay performance to estimate the available bandwidth.¹ Especially in wireless networks, a widely used approach is to measure the channel idle time and accordingly calculate a node’s available bandwidth. Previous works have reasoned that since nodes in the same neighborhood or in each other’s interference range share the same wireless channel, the total throughput of links interfering with each other along a path cannot exceed the channel bandwidth or the local available bandwidth. Based on this, they extend the channel idle time to path available bandwidth estimation.² Active probing on the other hand estimates the available bandwidth according to the characteristics of the received

¹State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China
²School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing, China

Corresponding author:
Tan Yang, School of Software Engineering, Beijing University of Posts and Telecommunications, No. 10, Xitucheng Road, Haidian District, Beijing 100876, China.
Email: tyang@bupt.edu.cn
The active probing bandwidth estimation tools further can be divided into two categories: the probe gap model (PGM) and the probe rate model (PRM).

PGM estimates the available bandwidth based on the time gap between the arrivals of two successive probes at the receiver. Assuming that a probe pair is sent with a time gap $t_r$ and reaches the receiver with a time gap $t_i$, the available bandwidth can be estimated by $t_r - t_i$, and link/path capacity. IGI/Ptr (initial gap increasing/packet transmission rate) and Delphi are typical algorithms using PGM.

PRM estimates the available bandwidth based on the probe rate between the sender and the receiver. In PRM, if the probe traffic is sent at a rate lower than the available bandwidth along the path/link, the arrival rate at the receiver will match with the sender rate. On the contrary, a higher sending rate results in queuing and delay of transmitting probing packets. Based on this, PRM measures the available bandwidth by identifying the turning point, where the probe sending and receiving rates match. Both Pathload and pathChirp belong to this category.

All above active algorithms are designed for wired networks. Recent research includes available bandwidth estimation algorithms specific to wireless networks, such as EXACT, ProbeGap, IdleGap, and DietTOPP. EXACT and IdleGap assume that the RTS/CTS (request to send/clear to send) is always enabled. EXACT captures the channel busy time and uses it to infer the available bandwidth, and IdleGap calculates available bandwidth using link idle rate and known capacity, and they both present only the simulation results. ProbeGap measures the available bandwidth indirectly from the idle time fraction using one-way delay samples, but it requires other path capacity estimation schemes. DietTOPP dynamically changes the bitrate of probing traffic to determine the available bandwidth when the probing traffic throughput experiences a turning point, which makes it have high intrusiveness in wireless network. Meanwhile, none of these algorithms addresses wireless layer dynamic rate adaptation. Wireless Bandwidth estimation tool (WBest) is a real-time estimation tool we know which is applicable to wireless networks with better performance. It belongs to the category of PGM technique and is a lightweight and fast available bandwidth technique. WBest employs a two-stage algorithm to determine available bandwidth along a path when the last hop is a wireless local area network (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. The evaluation of a wireless testbed shows WBest performs better in terms of accuracy, intrusiveness, and convergence time than the three other popular available bandwidth estimation tools: IGI/Ptr, pathChirp, and Pathload. Recently proposed NXET and NEXT-V2 determine the available bandwidth by analyzing the one-way queuing delay curve and detecting the turning point like pathChirp, and NEXT-V2 is an extended version of NEXT which enables to find the turning point more accurately. Different from pathChirp, they both have unique packet train structures, optimal rate adjustment algorithms, and modified excursion detection algorithms.

However, all of these active measurement tools need to be deployed on both the sender and receiver. They cannot work as long as one end of the end-to-end path does not have installation permission.

In wired network, some bandwidth estimation tools work based on round-trip-time (RTT) measurements, such as Clink and Pchar. However, Clink and Pchar are both designed for wired network, and they ignored that in wireless media for RTT measurements, the reply packet could conflict with the subsequent probing packet on the air interface. Besides, the convergence time of Clink and Pchar is too long, which makes them not applicable to wireless networks. Allbest is the first algorithm that successfully applies probe RTT measurements to estimate available bandwidth in WLANs. The core idea of Allbest is that the prober sends Maximum Transmission Unit (MTU)-sized User Datagram Protocol (UDP) packets to a non-activated port and then the receiver automatically replies Internet Control Message Protocol (ICMP) destination port unreachable packets. ICMP destination port unreachable packets are small and therefore they experience hardly any delay on the way back to the probing sender. But Internet Engineering Task Force (IETF) has extended ICMP slightly in RFC 4884. It basically says that ICMP destination port unreachable packet can now be of variable length. That change has been widely implemented in real-life stacks yet. Hence, for a MTU-sized UDP packet to a non-activated port, the reply ICMP error packet can be of hundreds of bytes. In addition, if the link/path capacity is symmetric, Allbest may also use ICMP echo request (Ping) probing packets and its echo reply packets to estimate available bandwidth. However, typically link/path capacity is not symmetric under wireless environment. Hence, the Allbest does not work in the real world now. In addition, other RTT-based algorithms have been proposed for wireless network. For example, GPing-Pair employs the variable packet size (VPS) probing technique to measure the end-to-end path capacity as a function of the packet size for cellular network, in which there is no interference between uplink and downlink signals. Therefore, GPing-Pair may not be applicable for 802.11 networks.
In this article, we propose a Round-trip Wireless Available Bandwidth estimation tool (RT-WABest), which employs a two-stage algorithm to estimate end-to-end available bandwidth in last hop WLANs. RT-WABest does not need a specific application running on the client like Allbest, and it improves Allbest and WBest: it can work in the real world by the round-trip way, in particular, it is also able to estimate the available bandwidth when the last link is not the bottleneck. In section “Algorithm,” we describe the RT-WABest algorithm. In section “Experiments,” we introduce our testbed in detail, followed by the experimental results and analysis in section “Results and analysis.” Our conclusions are given in section “Conclusion.”

Algorithm

The RT-WABest algorithm estimates available bandwidth on a network path where the last hop is over a wireless network, shown in Figure 1. In Figure 1, RT-WABest is deployed on an application server, and it first employs the packet-pair dispersion technique to estimate path capacity. Second, RT-WABest sends the client a packet train at the capacity rate to determine achievable throughput and then infers available bandwidth.

Capacity estimation

Generally, \( n \) packet pairs are sent back-to-back from the sender to the receiver. These packets are called probing packets of size \( L \). The path capacity can be calculated by

\[
C = \text{median}\left( \frac{L}{\text{OWD}_2 - \text{OWD}_1} \right) \quad (i = 1, 2, \ldots, n) \tag{1}
\]

where \( \text{OWD}_1 \) is the one-way delay of the first packet of \( i \)th packet pair, \( \text{OWD}_2 \) is the one-way delay of the second packet of \( i \)th packet pair, and \( \text{OWD}_2 - \text{OWD}_1 \) is the packet dispersion when the \( i \)th packet pair’s reply packets arrive at the sender.

In our RT-WABest, we apply probe RTT measurements to estimate capacity, which is given by

\[
C = \text{median}\left( \frac{L}{RTT_i^2 - RTT_i^1} \right) \quad (i = 1, 2, \ldots, n) \tag{2}
\]

where \( RTT_i^1 \) is the RTT of the first packet of \( i \)th packet pair, \( RTT_i^2 \) is the RTT of the second packet of \( i \)th packet pair, and \( RTT_i^2 - RTT_i^1 \) is the packet dispersion when the \( i \)th packet pair’s reply packets arrive at the sender.

RTTs of probing packets can be measured by sending MTU-sized Transmission Control Protocol (TCP) SYN packets to a non-activated port of the client side and then the client automatically generates TCP RST packets to the sender. TCP RST packets are very small and therefore experience hardly any delay on the way back to the probing sender, and the final result of equation (2) is then a good measure of capacity for the path from the sender to the client. But existing packet-pair dispersion techniques will not work in wireless media for RTT measurements, because the reply packet of the first probing packet could contend with the second probing packet on the air interface. Irrespective of which packet wins, the reply packet of the second probing packet will eventually arrive at the probing sender too late. As a result, capacity \( C \) will be underestimated.\(^{16}\)

To avoid the contention, Allbest sends a single packet with size of 2MTU, instead of two packets back-to-back. The 2MTU size packet will be fragmented and behaves like two individual packets back-to-back. And after defragmentation, only a single reply will be sent back by the client. Then, the \( RTT_2 \) will be calculated, which will not be delayed by additional contention. As the Allbest does not work in the real world now, to calculate the \( RTT_2 \) in equation (2), in our RT-WABest, we send a back-to-back packet pair. In this packet pair, the first packet is MTU-sized TCP RST packet to a non-activated port which will not be replied by the client, and the second packet is MTU-sized TCP SYN packet to a non-activated port. In this way, we can prevent the first packet’s reply packet from being put on the network while ensuring the second packet is replied. Besides, note that we can also calculate the correct \( RTT_1 \) in equation (2) by sending a series of separate single TCP SYN probing packets with size MTU.

Available bandwidth estimation

In WBest, the probing packets are sent at the rate of the path capacity \( C \) and then the available bandwidth \( A \) can be calculated from the average dispersion rate \( R \) at the receiver as\(^{12}\)

\[
A = \begin{cases} 
2C - \frac{C_i^2}{R}, & \text{if } R > \frac{C}{\tau} \\
0, & \text{else}
\end{cases} \tag{3}
\]

Note that WBest assumes last hop wireless network is the bottleneck link on the network path. If this
assumption does not hold, for example, an IEEE 802.11ac wireless network connecting with a lower capacity broadband Internet, shown in Figure 2, the probing traffic will share the same ratio of the total amount of traffic before and after reaching the bottleneck link at rate 

\[ \frac{C}{C + S} = \frac{R}{R + S'} = \frac{R}{C} \]  

(4)

where \( S \) is the crossing traffic or contending traffic on the bottleneck link before the probe packet arrives at this link, \( S' \) is crossing traffic’s share on the bottleneck link after the probe packet arrives. As we can obtain \( S = C^2/R - C \) from equation (4), the available bandwidth \( A = C - S \) is consistent with equation (3). In general, equation (3) can be applied to calculate the available bandwidth whether the last hop wireless link is the bottleneck of the path or not.

In the round trip measurement mode, RT-WABest also applies equation (3) for determining \( A \). It sends a packet train composed of \( M \) TCP SYN packets with size \( L \) to a non-activated port of the client side at the rate of the path capacity \( C \), and the client automatically generates TCP RST packets to the sender. TCP RST packets are very small and therefore experience hardly any delay on the way back to the probing sender. Then, the achievable throughput of the path from the sender to the client is given by

\[ R = \frac{L}{\text{mean}(T_i)} \quad (i = 1, 2, \ldots, M - 1) \]  

(5)

where \( T_i \) is the packet dispersion of reply TCP RST packets of the \( i \)th TCP SYN packet and the \( i + 1 \)th TCP SYN packet arriving at the sender. Then, based on capacity \( C \) and throughput \( R \), we can apply equation (3) for determining \( A \).

To avoid that the probing TCP SYN packet may contend with the reply TCP RST packet of previous probing TCP SYN packet in the last wireless hop and then the sending rate of packet train is slowed, RT-WABest does not send the next TCP SYN probing packet until the reply TCP RST packet of previous probing TCP SYN packet arrives. At the same time, the sender sets a threshold \( \tau \). And if the previous TCP SYN packet has not replied after \( \tau \) timeslots, the sender considers that the reply packet is lost and then continues to send the next TCP SYN packet immediately. Using the strategy above, in order to guarantee that the packet train speed reach the path capacity \( C \), we use the packet sending algorithm as follows. Note that \( L \) is the packet size, \( M \) is the number of packets in the packet train, and \( \tilde{T} \) is the length of timeslots.

Note that when \( N_i \) back-to-back packets were sent (line 25 of Algorithm 1), the first \( N_i - 1 \) packets are TCP ACK packets and the last one is TCP SYN packet. Then, only the last TCP SYN packet will be replied by the client; hence, the packet conflict does not occur. Meanwhile, when applying Algorithm 1, the \( T_i \) in equation (5) should be redefined as

\[ T_i = \begin{cases} T_i^{\text{rev}} - T_i^{\text{rev} - 1}, & \text{if } N_i \leq 1 \\ \frac{T_i^{\text{rev}} - T_i^{\text{rev} - 1}}{N_i}, & \text{if } N_i > 1 \end{cases} \]  

(6)

where \( i \) is consistent with the \( i \) in Algorithm 1.

Experiments

We evaluate RT-WABest in an IEEE 802.11 wireless testbed shown in Figure 3. The RT-WABest server is a PC with Intel Celeron 1.6 GHz CPUs and 2048 MB RAM which runs on Ubuntu 14.04 with Linux kernel version 3.13.0. The traffic generator is a PC with Intel Celeron 1.6 GHz CPUs and 2048 MB RAM running on CentOS with Linux kernel version 2.6.18. The client A is a laptop computer with Intel Core 2.5 GHz CPUs which runs on Ubuntu 14.04 with Linux kernel version 3.13.0. The client B is a laptop computer with Intel Core 2.5 GHz CPUs and 4096 MB RAM which runs on Ubuntu 12.04 with Linux kernel version 3.8.0. The...
servers connect to the access point (AP) with a wired 100 Mbps LAN, and the AP is a TP-LINK TL-WDR5600 with IEEE 802.11b/g mode or a TP-LINK TL-WDR7500 with IEEE 802.11ac mode.

In this testbed, probe traffic is sent by the RT-WABest server to the client A and then the client A replies the probes to the sender through the AP. Contending traffic comes from client B to the AP or from other clients and APs within interference range, accessing the shared wireless channel and competing with probing traffic on the path of interest.

We compare RT-WABest (https://github.com/Ppian/RT-WABest.git) with state-of-the-art wireless available bandwidth estimation tools, that is, WBest (http://perform.wpi.edu/downloads/wbest/wbest-1.0.tar) and Iperf (https://Iperf.fr/). In this testbed, WBest is installed on the RT-WABest server (acting as a sender) and the client A (acting as a receiver). According to the default setting of WBest, we choose 100 as the number of packet pairs, 60 as the length of the packet train, and 1500 bytes as the size of probe packet in Internet Protocol (IP) layer for WBest. And for RT-WABest, we choose 100 as the number of packet pairs, 400 as the length of packet train, and 1500 bytes as the size of probe packet after being encapsulated in the IP layer.

Note that Iperf could not estimate path capacity. To estimate available bandwidth, Iperf should send packets at a rate equal to or greater than the path capacity. Hence, we choose 11, 50, and 100 Mbps as the sending rate of Iperf in IEEE 802.11b, IEEE 802.11g, and IEEE 802.11ac mode, respectively.

The comparison should be performed in terms of accuracy, intrusiveness, and convergence time.19–21 And we commonly take these three factors into account and provide the results with figures and tables. Besides, we have designed more comprehensive cases to investigate the performance.

In different wireless access scenarios, that is, IEEE 802.11b/g and IEEE 802.11ac, we design five experimental cases, respectively, for evaluation. Case 1 has no crossing traffic and contending traffic, which means that the wireless channel is idle. In case 2, there is a crossing traffic flow from traffic generator to client B. In case 3, there is another crossing traffic flow from RT-WABest server to client B. In case 4, there is a contending traffic flow from client B to traffic generator. In case 5, there is a traffic flow from client B to

\textbf{Algorithm 1. Sending a packet train at rate C}

\begin{verbatim}
1: Procedure PacketTrainProber (L, C, M)
2: initialize i = 0; \(T = \frac{1}{C};\) timer = 0;
3: start the timer;
4: send the 1st TCP SYN packet at the beginning of the 1st timeslot, and record the send time \(T_{snd}^0;\)
5: \(\sigma_1 = 0;\) counter = 1;
6: repeat
7: \quad if \(\text{timer} < \tau \times \text{timeslots} \&\& \text{previous TCP SYN packet's reply TCP RST packet is not received}\)
8: \quad wait;
9: \quad else
10: \quad record the time \(T_{rev}^t;\)
11: \quad timer = 0;
12: \quad start the timer;
13: \quad i = i + 1;
14: \quad \tau_i = T_{rev}^{i-1} - T_{snd}^{i-1};
15: \quad N_i = \frac{\tau_i}{C_0};
16: \quad if N_i == 0
17: \quad send a TCP SYN packet at the beginning of the next timeslot;
18: \quad \sigma_{i+1} = 0;
19: \quad counter = counter + 1;
20: \quad else if N_i == 1
21: \quad send a TCP SYN packet immediately;
22: \quad \sigma_{i+1} = \tau_i + \sigma_i - T_i;
23: \quad counter = counter + 1;
24: \quad else
25: \quad send N_i back-to-back packets immediately, among them, the first \(N_i - 1\) packets are TCP RST packets, and the last one is TCP SYN packet.
26: \quad \sigma_{i+1} = \tau_i + \sigma_i - N_i\tau_i;
27: \quad counter = counter + N_i;
28: \quad end if
29: \quad end if
30: until counter = M
31: end procedure
\end{verbatim}
RT-WABest server, which acts as a contending traffic flow and a crossing traffic flow. In cases 2–5, we generated five different amounts of crossing traffic or contending traffic at different rates. In case 2 and case 4, there is only UDP traffic. In case 3, there is only Transmission Control Protocol (TCP) traffic. And in case 5, the background traffic is a combination of UDP traffic and TCP traffic. We used Iperf to generate UDP traffic and used the Multi-Generator Toolset (mgen) (http://pf.lit.nrl.navy.mil/mgen/) to generate TCP traffic with poissonian bitrate. Detailed experimental cases are shown in Table 1.

### Results and analysis

In each five cases with different wireless access scenarios, we run the three available bandwidth estimation tools 30 times, respectively, and list the median values in Figures 4–6. The measurements of case 1 are integrated into case 2–case 5.

#### Accuracy

**IEEE 802.11b/g access scenarios.** For 802.11b mode, the theoretical value of capacity in our testbed is 7.64 Mbps,\(^2\) that is, the theoretical value of available

### Table 1. Evaluation cases for experiments.

| Case | Mode     | Traffic (Mbps) | Cross | Contending |
|------|----------|----------------|-------|------------|
| 1    | IEEE 802.11b | 0              | 0     | 0          |
| 2 (UDP) | IEEE 802.11g | 1              | 0     | 2          |
| 3 (TCP) | IEEE 802.11ac | 0              | 5     | 0          |
| 4 (UDP) | IEEE 802.11g | 0              | 2     | 5          |
| 5 (UDP + TCP) | IEEE 802.11ac | 0.5 + 0.5  | 2.5 + 2.5 | 5 + 5  |

UDP: User Datagram Protocol; TCP: Transmission Control Protocol.
bandwidth in our testbed without background traffic is 7.64 Mbps. The measurements of RT-WABest, WBest, and Iperf are shown in Figure 4. It is clear that RT-WABest provides available bandwidth measurement results close to theoretical value for all cases. WBest provides larger measurement results than theoretical value when there is contending traffic (see case 4 and case 5). And Iperf performed badly when there is only TCP traffic (see case 3).

For 802.11b/g mode, the theoretical value of capacity in our testbed is 37.6 Mbps. The measurements of RT-WABest, WBest, and Iperf are summarized in Figure 5. Although RT-WABest tends to slightly underestimate available bandwidth in this mode (see case 2, case 3, and case 5), it still provides measurements closer to theoretical value than WBest and Iperf in most cases and provides accurate downward trend of available bandwidth. The reason of underestimation is that the reply TCP RST packets are delayed by the contending traffic. For all five cases, WBest begins to overestimate the available bandwidth when the rate of background traffic increases to 25 Mbps.

And Iperf provides puzzling measurements for all five cases.

From the above analysis, we can conclude that RT-WABest provides more accurate results than WBest and Iperf in both 802.11b mode and 802.11g mode. In 802.11b/g mode, WBest is more easily affected by the background traffic and tends to overestimate the available bandwidth especially when there is contending traffic. And Iperf is more easily affected by the TCP traffic. In addition, Iperf performs badly in 802.11g mode.

IEEE 802.11ac access scenarios. In all scenarios, we choose 1500 bytes as the size of probe packet in IP layer. In this situation, the encapsulation efficiency of Ethernet is 97.28%. Therefore, the theoretical value of capacity is 97.28 Mbps.

In Figure 6, RT-WABest provides available bandwidth measurements extremely close to the theoretical values for all cases. Because of high path capacity in this mode, the reply TCP RST packets are barely delayed by the contending traffic. So, RT-WABest can
provide a more accurate measurement of available bandwidth.

For all cases in Figure 6, available bandwidth provided by WBEST is value 0, which is because in the calculation of path capacity \( C \), most results are about 1946 Mbps, while the bottleneck rate in the testbed is not more than 100 Mbps. As achievable throughput of probes is not able to reach more than 100 Mbps, the results of available bandwidth will be 0 according to equation (3). Generally, significant overestimation of path capacity leads to value 0. And we consider that WBEST’s short convergence time (shown in Table 4) and probe packet package compression on the path lead to extremely large capacity measurements.

Iperf provides very accurate measurements in case 2 and case 3 except when there is high crossing traffic. However, in case 4 and case 5, Iperf seems not to have awareness of contending traffic and provides measurements close to the path capacity.

### Intrusiveness and convergence time

#### IEEE 802.11b/g access scenarios.

Table 2 lists the median of recorded intrusiveness and convergence time over 30 runs for the five cases in the 802.11b mode. As shown in Table 2, the convergence time of RT-WABEST is similar to WBEST’s. Although RT-WABEST provides a higher intrusiveness, the convergence time of RT-WABEST is so short that the intrusiveness can generally be ignored in real world. And in this mode, the intrusiveness of Iperf is 11 Mbps for all five cases.

#### IEEE 802.11ac access scenarios.

Table 4 lists the median of recorded intrusiveness and convergence time over 30 runs for the five cases in the 802.11ac mode. It is shown that RT-WABEST consumes less convergence time in most cases, which is one of the reasons that leads RT-WABEST to yield the higher intrusiveness. Another reason is that RT-WABEST needs to send more probe packets to ensure the accuracy of measurements. However, the intrusiveness can be generally ignored because of the short convergence time and the high capacity of IEEE 802.11ac mode. And in this mode, the
The intrusiveness of Iperf is 100 Mbps for all five cases. The results indicated that, no matter in which mode, Iperf is much slower and much more intrusive than WBest and RT-WABest.
Conclusion

In this article, we present an available bandwidth estimation tool for IEEE 802.11 wireless network, RT-WABest, designed based on RTT measurements with low intrusiveness and fast convergence time. RT-WABest is a two-stage algorithm; in the first stage, RT-WABest employs the packet-pair dispersion technique to estimate path capacity, and in the second stage, RT-WABest sends a packet train to infer available bandwidth. To the best of our knowledge, RT-WABest is the first applicable available bandwidth estimation tool based on RTT measurements for wireless networks.

In our wireless testbed, we compare RT-WABest with state-of-the-art tools, that is, WBest and Iperf. And the results show that, no matter in which mode, RT-WABest provides more accurate estimation in each evaluation case. In the future, we will implement RT-WABest under more complex wireless conditions and improve its accuracy and robustness.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the 863 Project of China (No. 2011AA01A102), the NSFC Project of China (No. 617271185), and the Fundamental Research Funds for the Central Universities of China (No. 2014RC0501).

References

1. Yuan Z, Venkataraman H and Muntean G. MBE: model-based available bandwidth estimation for IEEE 802.11 data communications. IEEE T Veh Technol 2012; 61(5): 2158–2171.
2. Chen F, Zhai H and Fang Y. Available bandwidth in multirate and multihop wireless ad hoc networks. IEEE J Sel Area Comm 2010; 28(3): 299–307.
3. Paul AK, Tachibana A and Hasegawa T. An enhanced available bandwidth estimation technique for an end-to-end network path. IEEE Trans Netw Serv Manag 2016; 13(4): 768–781.
4. Hu N and Steenkiste P. Evaluation and characterization of available bandwidth probing techniques. IEEE J Sel Area Comm 2003; 21(6): 879–894.
5. Ribeiro VJ, Coates MJ, Riedi RH, et al. Multifractal cross-traffic estimation. In: Proceedings of the ITC conference on IP traffic, modeling and management, Monterey, CA, 18–20 September 2000.
6. Jain M and Dovrolis C. Pathload: a measurement tool for end-to-end available bandwidth. In: Proceedings of the passive and active measurements (PAM) workshop, Fort Collins, CO, 24–26 March 2002.
7. Riberio VJ, Riedi RH, Baraniuk RG, et al. pathChirp: efficient available bandwidth estimation for network paths. In: Proceedings of the passive and active measurements (PAM) workshop, San Diego, CA, 6–8 April 2003.
8. Shah S, Chen K and Nahrstedt K. Available bandwidth estimation in IEEE 802.11-based wireless networks. In: Proceedings of the first ISMA/CAIDA workshop on bandwidth estimation (BEst), San Diego, CA, 9–10 December 2003.
9. Lakshminarayanan K, Padmanabhan VN and Padhye J. Bandwidth estimation in broadband access networks. In: Proceedings of the internet measurement conference (IMC), Taormina, 25–27 October 2004.
10. Lee HK, Hall V, Yum KH, et al. Bandwidth estimation in wireless LANs for multimedia streaming services. Adv MultiMed 2007; 2007: 70429-1–70429-7.
11. Johnsson A, Melander B and Bjorkman M. Bandwidth measurement in wireless networks. In: Proceedings of the Mediterranean ad hoc networking workshop, Porquerolles, 21–24 June 2005.
12. Li M, Claypool M and Kinicki R. WBest: a bandwidth estimation tool for IEEE 802.11 wireless networks. In: Proceedings of the IEEE local computer networks (LCN), Montreal, QC, Canada, 14–16 October 2008.
13. Chaudhari SS and Biradar RC. Survey of bandwidth estimation techniques in communication networks. Wireless Pers Commun 2015; 83(2): 1425–1476.
14. Paul AK, Tachibana A and Hasegawa T. NEXT: New enhanced available bandwidth measurement technique, algorithm and evaluation. In: Proceedings of the IEEE 25th annual international symposium on personal, indoor, and mobile radio communication (PIMRC), Washington, DC, 2–5 September 2014, pp.443–447. New York: IEEE.
15. Angrisani L, Botta A, Pescape A, et al. Measuring wireless links capacity. In: Proceedings of the IEEE 1st international symposium on wireless pervasive computing, Phuket, Thailand, 16–18 January 2006, pp.1–5. New York: IEEE.
16. Delphinanto A, Koonen T and Hartog F. End-to-end available bandwidth probing in heterogeneous IP home networks. In: Proceedings of the 8th annual IEEE consumer communications and networking conference: multimedia & entertainment networking and services, Las Vegas, NV, 9–12 January 2011. New York: IEEE.
17. https://tools.ietf.org/html/rfc4884
18. Zhong M, Hu P and Jadwiga I. Revisited: bandwidth estimation methods for mobile networks. In: Proceedings of the IEEE 15th international symposium on world of wireless, mobile and multimedia networks (WoWMoM), Sydney, NSW, Australia, 16–19 June 2014. New York: IEEE.
19. Goldoni E and Schivi M. End-to-end available bandwidth estimation tools, an experimental comparison. In: Ricciato F, Mellia M and Biersack E (eds) Traffic monitoring and analysis. Berlin, Heidelberg: Springer, 2010, pp.171–182.
20. Shriram A, Murray M, Hyun Y, et al. Comparison of public end-to-end bandwidth estimation tools on high-speed links. In: Shriram A, Murray M, Hyun Y, et al. (eds) Passive and active network measurement. Berlin, Heidelberg: Springer, 2005, pp.306–320.

21. Prasad R, Dovrolis C, Murray M, et al. Bandwidth estimation: metrics, measurement techniques, and tools. IEEE Network 2003; 17(6): 27–35.

22. Delphinanto A, Koonen T, Zhang S, et al. Path capacity estimation in heterogeneous, best-effort, small-scale IP networks. In: Proceedings of the 35th IEEE conference on local computer networks (LCN 2010), Denver, CO, 10–14 October 2010. New York: IEEE.