An Improved Available Bandwidth Measurement Algorithm based on Pathload

LIANG ZHAO* CHONGQUAN ZHONG
Faculty of Electronic Information and Electrical Engineering
Dalian University of Technology
Dalian 116024
China

Email:zliangdut@qq.com

Abstract: - Available network bandwidth is an important metrics of the network performance. Quickly and accurately measuring the available network bandwidth is significant to network monitoring, congestion control, and network traffic projects. The traditional Pathload scheme for measuring the available network bandwidth suffers heavy measurement overheads and slow convergence. Based on the analysis of Pathload, a new method is proposed to improve Pathload's decision criterion and the sending rate adjusting algorithm. NS2 simulations show that our method increases the measurement accuracy, reduces measurement overheads and speeds up convergence.

Key-Words: - Bandwidth Measurement; Network Available Bandwidth; Simulation; Pathload

1 Introduction

Network bandwidth is the largest number of data bits that can be sent across the links in a unit of time, and available network bandwidth is the end-to-end maximum data transmission rate without affecting the background traffics along the network paths [1]. Their units are both bit/s. For an end-to-end network transmission system, the available bandwidth is of greater practical value, because it is crucial to balancing network loads, controlling transmission rates and network congestion. Let C denote the link bandwidth in bit/s, u denote the bandwidth utilization in bit/s, H denote the number of end-to-end link hops, and A denote the currently available bandwidth in bit/s, then we have:

$$C = \min_{i=1\cdots H} C_i \tag{1}$$

$$A_i = C_i\left(1-u_i\right) \tag{2}$$

$$A = \min_{i=1\cdots H} A_i = \min_{i=1\cdots H}\left\{C_i\left(1-u_i\right)\right\} \tag{3}$$

The existing methods for measuring the end-to-end available bandwidth can be categorized into two man classes: single-end measuring and dual-end measuring [2]. In the single-end scheme, the measuring software is installed only on one end to measure the available link bandwidth. But in the dual-end mode, the measuring software is installed on both the client and the server. On comparison, the single-end scheme is easy to implement but its measuring accuracy is lower than that of the dual-end measuring system. Table 1 lists the common methods for measuring available network bandwidth, where the single-end methods include: Cprobe[3]、Pipechar[2]、SProbe[4]、Abget[5]、ICMP-SLoPS[6], and the dual-end methods include: Delphi[7]，Pathload[1, 8]，Pathchirp[9]，IGI[10]，Spruce[11]，TOPP[12].

Tab. 1 Measurement methods for network available bandwith

Classification	Name	Protocol	Year
Single side measurement	Cprobe	ICMP	1996
	Pipechar	ICMP	2001
	SProbe	TCP	2002
	Abget	TCP	2005
	ICMP-SLoPS	ICMP	2008
	Delphi	UDP	2000
Double Sides measurement	Pathload	TCP	2001
	Pathchirp	TCP	2002
	IGI	TCP	2003
	Spruce	TCP	2004
	TOPP	TCP	2005

No method above is more accurate than Pathload, but it measures the data flow using the self-congestion strategy, resulting in heavy time consumption and overheads. Over ten seconds are needed to measure the bandwidth once using this method. After about 1MB probe packets are injected into the network, it is difficult to ensure the network traffics remain unchanged during measurement. Pathchirp only takes seconds to measure the bandwidth once, but it also needs to inject about 200kb probe data into the network.

In addition, the above methods are based on the following 2 assumptions:

① Network loads form a FIFO (first input first output) queue at the routing nodes.

② The network is a fluid model, and the background traffic is unchanged during the measurement period.

2 Principles for Pathload Measuring Available Bandwidth

Pathload is based on SloPS (self-loading periodic stream) to measure the effective bandwidth of the end-to-end links. The periodic probe stream is sent at a rate of R from the sender to the receiver. The relation between the periodic stream sending rate and the end-to-end link effective bandwidth is determined according to the distribution of the packet one-way delay (OWD) in arriving at the receiver. The sending rate of the probe streams is adjusted via dichotomy to approximate to the effective link bandwidth.

Definition [1]**:** if the sending rate R > the available bandwidth A, then the OWD difference $\Delta D^k > 0$, showing an upward trend. If the sending rate R < the available bandwidth A, then $\Delta D^k \approx 0$ or the trend is not obvious.

Let R_0 denote the sending rate of the first packet, C_1 denote the bandwidth of the first routing node, A_1 denote the available bandwidth capacity of the first routing node, and u_1 denote the bandwidth utilization of the first router.

If $R_0 > A_1$, then

$$A_1 = (1-u_1)C_1 = C_1 - u_1 C_1 \quad (4)$$

$$R_0 + u_1 C_1 = R_0 + (C_1 - A_1) = C_1 + (R_0 - A_1) > C_1 \quad (5)$$

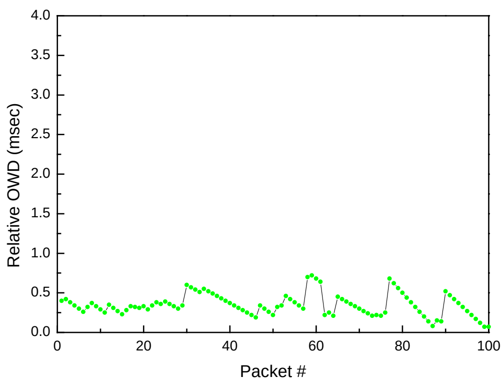
Equation (5) shows that packet backlogs have occurred at the first routing node, i.e. the queuing delay arises. Let t^k be the arriving time of the k^{th} packet, $T = \frac{L}{R_0}$, then the routing queue at the time $\left[t^k, t^k + T\right]$ is:

$$\begin{aligned}\Delta q_1^k &= (L + u_1 C_1 T) - C_1 T \\ &= R_0 T - (1-u_1) C_1 T = (R_0 - A_1) T > 0\end{aligned} \quad (6)$$

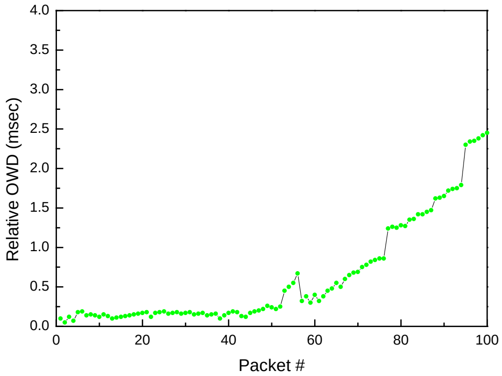
Queuing delay difference

$$\Delta d_1^k = \frac{R_0 - A_1}{C_1} T > 0 \quad (7)$$

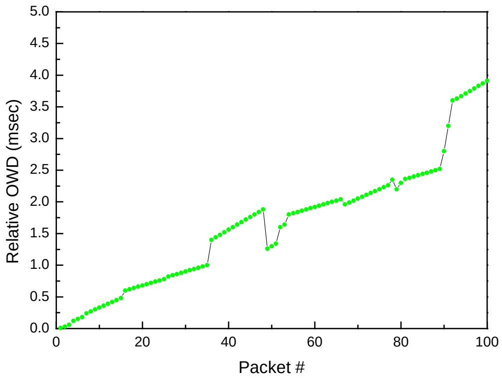
Similarly, it can be proven that when the sending rate of the probe stream R < the available bandwidth A, then the OWD different does not show an obvious upward trend. Further illustrations are given in Figures 1 (a), (b) and (c), where the sender sends 100 probe packets each time to the receiver at a constant rate of R, and it has three cases: $R_a<A$, $R_b \approx A$, $R_c>A$. From Figure 1(a), it is obvious that when the sending rate $R_a<A$, the delay fluctuates in a range instead of visibly showing an upward or downward trend. When $R_c>A$, the delay increases overall, despite few fluctuating or decreasing points, as shown in Figure 1(c). In Figure 1(b), the delay of the first half probe stream fluctuates, meaning that $R_b<A$; the delay of the second half increases, meaning that the available bandwidth begins to decrease and is smaller than the sending rate. This leads to an estimate of the effective bandwidth $A \approx R_b$.



（a）Ra<A



（b）Rb≈A



（c）Rc>A

Fig. 1 One Way Delay change curve

3 Pathload Decision Criterion

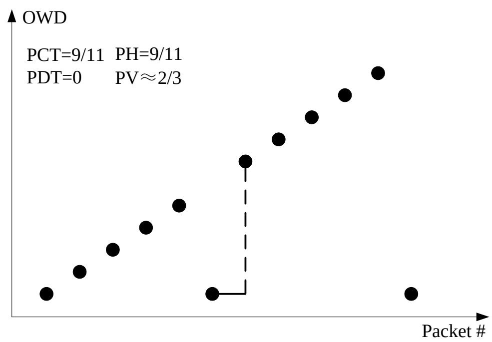
Pathload uses parameters SPCT and SPDT to determine whether OWD increases [1, 8], as shown in Equation (8).

$$\begin{cases} S_{PCT} = \dfrac{\sum_{i=2}^{\Gamma} I\left(\hat{D}_i > \hat{D}_{i-1}\right)}{\Gamma - 1} \\ S_{PDT} = \dfrac{\hat{D}_{\Gamma} - \hat{D}_1}{\sum_{i=2}^{\Gamma} \left|\hat{D}_i - \hat{D}_{i-1}\right|} \end{cases} \quad (8)$$

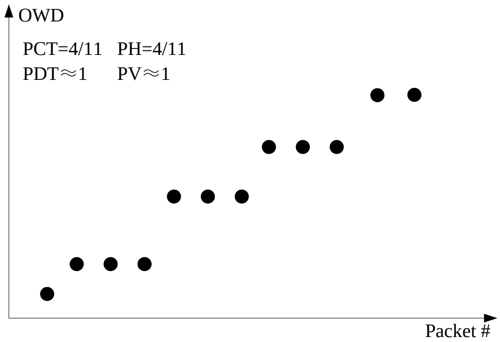
Where K denotes the length of the data stream (typically value being 100). To avoid the effect of few radically fluctuating delays, the filtering factor Γ ($\Gamma = \sqrt{K}$) is introduced to segment the original data stream. Let $\hat{D}_i\left(i = 1, \cdots, \Gamma\right)$ denote the average delay of each segment. If the condition x holds true, then $I\left(x\right)$ is equal to 1; otherwise, it is equal to 0. Obviously, S_{PCT} represents the probability that the two neighboring points along the abscissa axis shows an upward trend. S_{PDT} represents the proportion of the first point subtracted from the last point along the longitudinal axis to the delay sum of two neighboring points. Jain reported in [8] that the delay may fluctuate randomly, so after being filtered, the delay of the last segment may be comparable to the delay of the first segment. Therefore, these two parameters need to be used together to check whether the delay shows an upward trend.

Here, the delay distribution in [8] is supplemented, as shown in Figure 2, where Figures 2(a) and (b) are from [8], whereas Figures (c) and (d) are newly added in this paper. It can be seen that when the first and last sets of delays fluctuate, S_{PCT} and S_{PDT} cannot accurately describe the upward trend of the delay. In this context, S_{PDT} is improved by replacing the last point with the maximum delay $\hat{D}_{\max}$, and the first point with the minimum delay $\hat{D}_{\min}$. Meanwhile, according to the physical meanings of the decision parameters, S_{PCT} is defined as S_{PH} (Probability in Horizontal), and S_{PDT} is defined as S_{PV} (Proportion in Vertical), as shown in Equation (9). That is, whether the delay increases in the horizontal and vertical directions is determined separately, and the two directions are complementary to each other. Figures 2(c) and (d) show that when S_{PDT} fails, S_{PV} can properly explain the delay's upward trend.

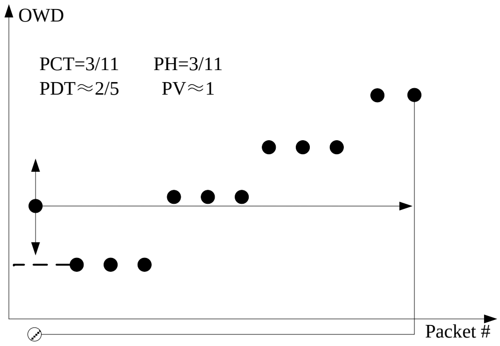
$$\begin{cases} S_{PH} = \dfrac{\sum_{i=2}^{\Gamma} I\left(\hat{D}_i > \hat{D}_{i-1}\right)}{\Gamma - 1} \\ S_{PV} = \dfrac{\hat{D}_{\max} - \hat{D}_{\min}}{\sum_{i=2}^{\Gamma} I\left(\hat{D}_i - \hat{D}_{i-1}\right)} \end{cases} \quad (9)$$



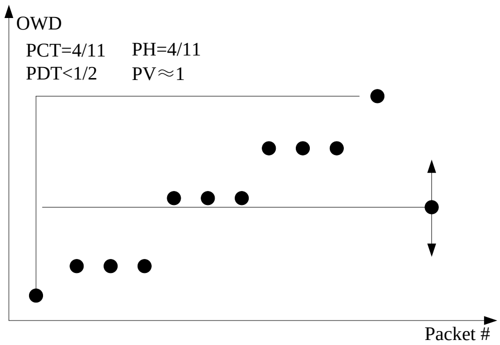
（a）



(b)



(c)



(d)

Fig. 2 One Way Delay statistical distribution

According to the definitions in (9), 0<S_{PH}<1, 0<S_{PV}<1. Based on the thresholds in [8], we propose the combined decision criterion for the upward trend. When S_{PH} >0.66 or S_{PV} >0.45, it is decided that the delay shows an obvious upward trend. When S_{PH} <0.54 or S_{PV} <0.45, it is decided that the delay shows no upward trend. When S_{PH} <0.54 or S_{PV} <0.45, it is decided that the trend of the delay is unknown. The combined decision criterion for the upward trend is detailed in Table 2.

Tab. 2 Discriminance method for rising trend of time delay

	S_{PH} <0.54	0.54<S_{PH} <0.66	S_{PH} >0.66
S_{PV} <0.45	No trend	No trend	unable to determine
0.45<S_{PV}<0.55	No trend	unable to determine	rising trend
S_{PV} >0.45	unable to determine	rising trend	rising trend

4 Improved Available Bandwidth Measuring Algorithm

According to the Pathload principles for measuring available bandwidth, the SLoPS-based rate adjusting algorithm computes the sending rate of the next time via dichotomy to approximate it to the end-to-end available bandwidth. After initializing $G^{\min} = G^{\max} = 0$, the details of the rate adjusting algorithm are as follows:

(1) The delay shows no upward trend

$$\begin{array}{l} if\left(R(n) < A\right) \\ \quad R^{\min} = R(n); \\ \quad if\left(G^{\min} > 0\right) \\ \qquad R(n+1) = \dfrac{G^{\min} + R^{\min}}{2}; \\ \quad else \\ \qquad R(n+1) = \dfrac{R^{\max} + R^{\min}}{2}; \end{array}$$

(2) The delay shows an upward trend

$$\begin{array}{l} if\left(R(n) \geq A\right) \\ \quad R^{\max} = R(n); \\ \quad if\left(G^{\max} > 0\right) \\ \qquad R(n+1) = \dfrac{G^{\max} + R^{\max}}{2}; \\ \quad else \\ \qquad R(n+1) = \dfrac{R^{\max} + R^{\min}}{2}; \end{array}$$

(3) The delay is distributed in the gray area

$$
\begin{aligned}
& if\left(G^{\min} == G^{\max} == 0\right) \\
& \quad G^{\min} = G^{\max} = R(n); \\
& if\left(G^{\max} \le R(n)\right) \\
& \quad G^{\max} = R(n); \\
& \quad R(n+1) = \frac{G^{\max} + R^{\max}}{2}; \\
& else\,if\left(G^{\min} > R(n)\right) \\
& \quad G^{\min} = R(n); \\
& \quad R(n+1) = \frac{G^{\min} + R^{\min}}{2};
\end{aligned}
$$

The slow convergence of the SLoPS rate adjusting algorithm directly leads to the heavy time consumptions and bandwidth overheads in Pathload. The Pathload algorithm will be more feasible if the rate adjusting algorithm's convergence can be sped up.

According to (7), the difference ΔT between the time when the *k+1*th packet leaves the first server and the time when the *k*th packet leaves the first server is:

$$
\begin{aligned}
\Delta T &= \left(t^{k+1} + d_1^{k+1}\right) - \left(t^k + d_1^k\right) \\
&= T + \frac{R_0 - A_1}{C_1} T
\end{aligned}
\tag{10}
$$

It can be seen that the time when two consecutive packets leave the server is independent of the number of packets k. That is, the first set of data streams leaves the first server at a constant rate:

$$
\begin{aligned}
R_1 &= \frac{L}{\Delta T} = \frac{L}{T} \frac{C_1}{C_1 + \left(R_0 - A_1\right)} \\
&= R_0 \frac{C_1}{C_1 + \left(R_0 - A_1\right)}
\end{aligned}
\tag{11}
$$

It can be inferred that the relation between the rate R_{i-1} at which the segmented data streams arrive at the i^{th} server, the rate R_i at which the segmented data streams leaves the i^{th} server, and the available bandwidth A_i at this node, satisfies the following equation:

$$
\begin{cases}
A_i \le R_i \le R_{i-1} & R_{i-1} > A_i \\
R_i = R_{i-1} & R_{i-1} \le A_i
\end{cases}
\tag{12}
$$

Observe the codes of the rate adjusting algorithm that indicate an upward trend of the delay, in the case of $R(n) \ge A$, if the rate $R^*(n)$ at which the segmented data leaves the server is used as an replacement, i.e. $R^{\max} = R^*(n) \quad \left(if\left(R(n) \ge A\right)\right)$, then the upper limit of the rate can be reduced substantially, narrowing its difference with the available bandwidth, accelerating the algorithm convergence, lowering the algorithm's measurement overheads.

5 NS2 Simulations

In this section, the simulator NS2 is used to evaluate the performance of the improved decision criterion and the Pathload algorithm. The end-to-end network topology designed for the experiment in [14] is shown in Figure 3. There are five nodes between the client and the server, the R3-R4 link is the tight link with a bandwidth capacity of 10Mb/s, and links between other nodes are normal links with a capacity of 20Mb/s. The background traffics of the network are generated by the source following the self-similar Pareto distribution, where $\alpha = 1.9$, the 550B packets account for 50%, 40B packets account for 40% and 1500B packets account for the rest 10%.

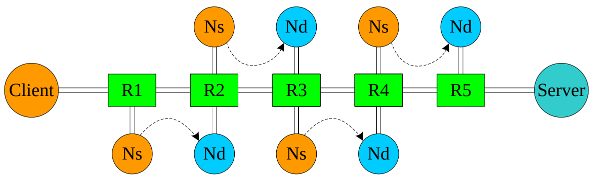


Fig. 3 Simulation topology figure of End-to-End network

The links are classified according to the network loads into four categories: $u_1 = 20\%$, $u_2 = 40\%$, $u_3 = 60\%$, and $u_4 = 80\%$, whose actual effective bandwidth corresponds to 8Mb/s, 6Mb/s, 4Mb/s and 2Mb/s, respectively. The length of the probe packet in the experiment is set to 300B, the accuracy parameter is set to ω =1Mb/s, and χ =0.2Mb/s. If the algorithm has not converged after sending 10 probe packets, then the probing procedures must be terminated, generating an optimal estimate based on the lower and upper limits of the currently available bandwidth.

Instead of directly providing an estimate of the available network bandwidth, the original measuring method offers an interval consisting of the lower and upper limits through the measuring processes. Here, the mean of the lower and upper limits are computed as the final measurement result. In addition, the gateway of the data collector can use this value to adjust the sending rate. The experimental results are given in Table 3. By updating the lower and upper limits of the delay, the improved method narrows the interval of the measurement result, making the acquired available bandwidth closer to the actual available bandwidth. Furthermore, the convergence is sped up, reducing the measurement overheads and the negative impacts on the network.

Tab. 3 Comparison of network available bandwidth measurement accuracy

bandwidth availability ratio	Real bandwidth Mb/s	Original method		Improved method	
		Measuring Results Mb/s	Network flow Mb	Measuring Results Mb/s	Network flow Mb
u_1=20%	8	2.24	13.44	1.99	6.72
u_2=40%	6	4.47	13.19	4.23	7.41
u_3=60%	4	6.5	12.78	6.31	8.72
u_4=80%	2	8.31	11.52	8.18	9.6

6 Conclusion

The Pathload scheme for measuring available network bandwidth suffers heavy measurement overheads and slow convergence. In this context, improvements are made to the decision criterion and the adjustment of the sending rate. By updating the upper and lower limits of the available bandwidth more quickly, the proposed method speeds up its convergence and reduces the measurement overheads. The NS2 simulations prove the effectiveness of the proposed method.

6 Acknowledgments

The present research is supported by the Dalian University of Technology Scientific research start-up funding (DUT14RC(3)129).

References:

[1] Jain M, Dovrolis C. End-to-End available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput [J].IEEE/ACM Transactions on Networking, 2003,11 (4):537-549.

[2] Zhou Hui, Li Dan, Wang Yongji. Fundamental problems with available bandwidth measurement systems [J]. 2008,(05):1234-1255.

[3] Carter R L, Crovella M E. Measuring bottleneck link speed in packet-switched networks [J].Performance Evaluation, 1996,27–28:297-318.

[4] Saroiu S, Gummadi P K, Gribble S D. Sprobe: A fast technique for measuring bottleneck bandwidth in uncooperative environments[C].Proc. of the IEEE INFOCOM, IEEE,2002.

[5] Antoniades D, Athanatos M, Papadogiannakis A, et al. Available bandwidth measurement as simple as running wget[C].Proceedings of the Passive and Active Measurement Conference, 2006.

[6] Zha Xintian, Xu Xiaodong. Available bandwidth measurement with ICMP in IPv6 networks [J]. Microcomputer Information. 2008,(3):121-123.

[7] Ribeiro V J, Coates M J, Riedi R H, et al. Multifractal Cross-Traffic estimation[C].Proc ITC Specialist Seminar on IP traffic Measurement, Modeling, and Management, Monterey, California: 2000.

[8] Jain M, Dovrolis C. Pathload: A measurement tool for End-to-End available bandwidth[C].In Proceedings of Passive and Active Measurements (PAM) Workshop, Fort Collins: 2002.

[9] Ribeiro V J, Riedi R H, Baraniuk R G, et al. PathChirp: Efficient available bandwidth estimation for network paths[C].Work shop on Passive and Active Measurement (PAM), La Jolla, California: 2003.

[10] Hu N, Steenkiste P. Evaluation and characterization of available bandwidth probing techniques [J].IEEE Journal on Selected Areas in Communications, 2003,21 (6):879-894.

[11] Strauss J, Katabi D, Kaashoek F. A measurement study of available bandwidth estimation tools[C].Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement, New York: ACM,2003.

[12] Melander B, Bjorkman M, Gunningberg P. A new end-to-end probing and analysis method for estimating bandwidth bottlenecks[C].Global Telecommunications Conference, IEEE,2000.