Improving Accuracy of Bandwidth Estimation for Internet Links by Statistical Methods

Kazumine Matoba
Osaka University, Japan
k-matoba@ics.es.osaka-u.ac.jp

Shingo Ata
Osaka City University, Japan
ata@info.eng.osaka-cu.ac.jp

Masayuki Murata
Osaka University, Japan
murata@ics.es.osaka-u.ac.jp

Abstract

Network dimensioning is an important issue to provide stable and QoS–rich communication services. A reliable estimation of bandwidths of links between the end–to–end path is a first step towards the network dimensioning. `Pathchar` *is one of such tools for the bandwidth estimation for every link between two end hosts. However,* `pathchar` *still has several problems. If unexpectedly large errors are included or if route alternation is present during the measurement, the obtained estimation is much far from the correct one. We investigate the method to eliminate those errors in estimating the bandwidth. To increase the reliability on the estimation, the confidence interval for the estimated bandwidth is important. For this purpose, two approaches, parametric and nonparametric approaches, are investigated to add the confidence intervals. Another important issue is the method for controlling the measurement period. If the link is stable, small measurement data is sufficient. On the other hand, if the data is not sufficient, many measurements is necessary to obtain an accurate and reliable estimation. In this paper, we propose a measurement method to adaptively control the number of measurement data sets.*

1. Introduction

Network dimensioning is becoming a more and more important issue of the day in the Internet. Stable and QoS–rich communication services cannot be provided unless the network is properly dimensioned. One typical example can be found in a diff–serv architecture [1] where the bandwidth should be adequately prepared for QoS classes. Another example is MPLS [2] and IP–over–WDM networks. In such a network architecture, the physical path capacity should be determined a priori.

However, in the Internet, it is difficult to know or to estimate the traffic demand in advance mainly due to the following two reasons. The one is that the Internet is growing drastically and therefore it cannot forecast future demands of user traffic. The other is owing to the characteristics of the Internet traffic. A dominant of the Internet traffic is TCP–based application having a capability of adapting to network congestion. It suggests that the network monitoring should be performed not only at the node and/or link but also in an end–to–end fashion. Accordingly, various tools have been developed to measure the traffic characteristics on the Internet. See, e.g., [3].

An accurate and reliable estimation of the bandwidth of links on the end–to–end path is a first step towards network dimensioning. `Pathchar` [4, 5] is one of such tools to measure latency, bandwidth, queueing delays and packet loss rate for every link between two hosts. The advantage of `pathchar` is that it is not necessary to deploy new protocols with any special functions at both of routers and end hosts. `Pathchar` collects RTTs (Round Trip Time) with various sizes of packets and estimates the link bandwidth according to the relation of RTTs and packet sizes.

However, `pathchar` still has several problems as we will explain in detail in Section 2. In short, `pathchar` needs a large amount of statistics to improve the bandwidth estimation, which is obtained by throwing the large number of packets into the network. However, it has an intrinsic problem that the increased traffic may cause congestion and an estimated value is biased by `pathchar` itself. Instead of pursuing the accuracy of the approach taken by `pathchar`, we take another approach to add a confidence in the estimation. A recent version of `patchar`, which is now called as `pchar`, gives a confidence interval for the slope (by which the bandwidth estimation is derived), but it is insufficient for the user to rely on the obtained results. In this paper, we investigate the calculation method to determine the confidence intervals for the estimated bandwidth. The control method for measurement time is also proposed to limit the unnecessary probes injected into the network.

The rest of this paper is organized as follows. Section 2 briefly introduces `pathchar` and point out several problems that we want to resolve. In Section 3, we propose our estimation method of the link bandwidths with confidence intervals. In Section 4, experimental results of our measurement method are shown. We conclude our paper with future research topics in Section 5.

2. A Brief Description on Pathchar and its Problems

2.1. A Brief Description on Pathchar

In this subsection, we summarize a bandwidth estimation method taken in pathchar. For more details, refer to [4].

Pathchar first collects RTTs between source and destination hosts. To measure RTTs, pathchar uses one of the ICMP packet, called a *Time Exceeded* message, which is also used in traceroute [6]. An IP packet has a TTL (Time To Live) field in the header. It shows the limit of the hop count that the packet can traverse. Before the router forwards the packet to the next hop, the value of the TTL field is decreased. When the TTL value becomes zero, the router discards the packet and returns the ICMP control packet to the source to inform that the validity of the packet is expired. This mechanism is necessary in order to avoid the loops of packet forwarding due to, e.g, some misbehavior of the router. When the packet is sent with the value of the TTL field to be n, the ICMP control packet must be returned from nth hop router. The RTT value between the source and nth router on the path can then be measured by the source. Pathchar collects RTTs between the source and every intermediate router by changing the value of the TTL field.

The measured RTT value consists of (1) the sum of queueing delays, q_i, at router i $(1 \le i \le n)$, (2) the sum of transmission times to transmit the packet by the intermediate routers, (3) the sum of forwarding times f_i that router i processes the packet, and (4) the sum of propagation delays p_j of link j $(1 \le j \le n)$. That is, RTT_s, the RTT value for given packet size s, is represented by

$$RTT_s = \sum_{j=1}^{n} \left(\frac{s}{b_j} + \frac{s_{ICMP}}{b_j} \right) + \sum_{i=1}^{n} (q_i + f_i) + 2 \sum_{j=1}^{n} p_j . \quad (1)$$

where s_{ICMP} is a size of an ICMP error message and b_j is the bandwidth of link j.

A typical example for the relation between packet sizes and measured RTTs is shown in Figure 1. The results are obtained by setting the destination to be www.gulf.or.jp from our site. The TTL value was set to 16. It was obtained on Dec 18, 1999 12:54 JST. The figure shows that the RTT values were widely spread even for the fixed packet size. It is because the queueing delay at the router changes frequently by the network condition. However, it is likely that several packets do not experience the queueing delays at any router by increasing the trials. Such

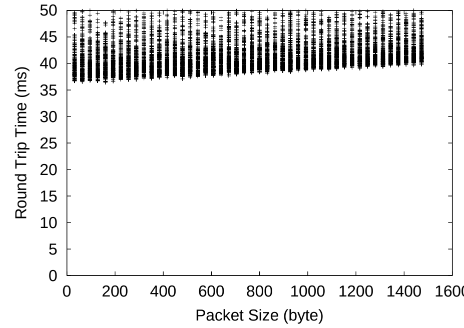

Figure 1: Distribution of RTT values vs. packet size

a case actually appears in the figure as a minimum value of RTTs for each packet size. The minimum RTT for given packet size s, denoted by $minRTT_s$, is thus obtained by

$$minRTT_s = \sum_{j=1}^{n} \frac{s + s_{ICMP}}{b_j} + \sum_{i=1}^{n} f_i + 2 \sum_{j=1}^{n} p_j . \quad (2)$$

Note that the packet size of the ICMP error message s_{ICMP} is fixed (56 bytes). Then, by collecting terms not related to the packet size and denoting it by α, the above equation can be rewritten as

$$minRTT_s = s \sum_{j=1}^{n} \frac{1}{b_j} + \alpha. \quad (3)$$

Eq. (3) is a linear equation with respect to the packet size s. It is just shown in Figure 1 if we look at the minimum RTT values. By letting the coefficient of the above equation be β_n, we have

$$\beta_n = \sum_{j=1}^{n} \frac{1}{b_j}. \quad (4)$$

Conversely, if we have β_{n-1} and β_n, we can obtain the bandwidth of link j as

$$b_j = \frac{1}{\beta_n - \beta_{n-1}}. \quad (5)$$

It is a key idea of pathchar.

As indicated above, a difficulty of pathchar exists in that in real networks such as the Internet, the network condition changes frequently, and it is not easy to obtain proper minimum RTTs. Thus, pathchar needs to send many packets with the same size; it is a weak point of pathchar since those waste a large amount of link bandwidth to get a minimum RTT. Even after many RTTs are collected, some measurement errors must be contained. Pathchar solves this problem by a linear least square approximation.

2.2. Problems of Pathchar

The approach of the bandwidth estimation taken by pathchar is innovative, but it still has several problems as described below.

2.2.1 . Reliability on Obtained Estimation

First, we cannot know whether the estimated bandwidth obtained by pathchar is reliable or not. Pathchar uses the linear least square fitting to calculate β_n, which means that it assumes errors of minimum RTTs are normally distributed. However, we have no means to confirm whether errors follow a normal distribution or not. From this reason, it is necessary to consider another approach that can lead to bandwidth estimation independently of the error distribution. Such an approach is often called as a nonparametric approach. The nonparametric approach is already developed in pchar [7], an updated version of pathchar. While in pchar the user can choose the parametric or the nonparametric method for estimation, it does not offer any criterion to decide which approach is better. Another nonparametric approach is proposed in [4]. In this paper, he proposed an original approach to control the number of measurement packets but we should use generic statistical estimate method.

2.2.2 . Efficiency of Measurements

The second problem is the efficiency of pathchar. Pathchar sends a fixed number of packets, but the amount of collected data must be changed according to the network condition to measure the link bandwidth within a reasonable level of accuracy. The author in [4] then propose an *adaptive* data collection method to improve the efficiency of pathchar. They have shown that the required number of packets in pathchar can much be reduced if pathchar is equipped with an ability to send a different number of packets for each link estimation. In their proposal, the number of transmitted packets is decided by observing whether the even-odd range of bandwidth is converged or not. However, the range is not based on the reliability on the result and the method does not guarantee an accuracy in a *statistical* sense. We should apply the confidence interval which is based on the statistics.

2.2.3 . Exceptional Errors of RTTs

The third problem is that various kinds of errors are mixedly contained in minimum values of RTT. Nevertheless, pathchar assumes that the error of the minimum RTT is originated from the measurement noise only, and assumes the normal distribution for measurement errors. Basically, pathchar relies on the fact that the queueing delay at the intermediate routers can be removed by gathering a number of measurements since one or more packets must fortunately encounter no queueing delay by increasing the measurement. If the number of measurements is insufficient, the queueing delay may be involved. However, it may be able to be viewed as a Gaussian noise.

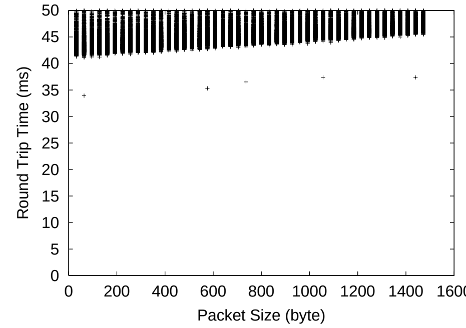

Figure 2: A Sample not following a normal distribution

The problem is that we encounter the errors which cannot be explained by the Gaussian noise. One example is shown in Figure 2, which was obtained at Dec 21, 1999 08:39 JST by setting the destination as www.try-net.or.jp and the TTL value as 13. Several small values were observed during the measurement as shown in Figure 2. We need to introduce some method to remove such errors before the bandwidth estimation is performed. For this purpose, we will apply a weighted least square fitting method as will be explained in Subsection 3.1.

A second example was obtained by route alternation. To get the bandwidth estimation, all probes should be relayed on the same path. If pathchar detects the route changes by checking the field of the source IP address in the returned ICMP packet, it simply discards the returned packet. Such a case may happen due to load balancing at routers [8]. The problem is that it cannot eliminate the case where the source IP addresses of the returned ICMP packets are same, but the relayed paths are different. In fact, we obtained such measurement as shown in Figure 3. It was observed at 8-th link destined for www.kyotoinet.or.jp at Dec 10, 1999 12:29 JST. Figure 3 clearly shows that there exist two (or maybe more) paths during the measurement. To remove such an effect, we need to select the proper subgroup of RTTs for accurate estimation, which will be explained in Subsection 3.2

3. Accuracy and Reliability Improvements for Bandwidth Estimation

As we have discussed in the previous section, we need to solve several problems for obtaining accurate and reliable bandwidth estimation. For this purpose, we first examine two estimation methods; parametric and nonparametric ap-

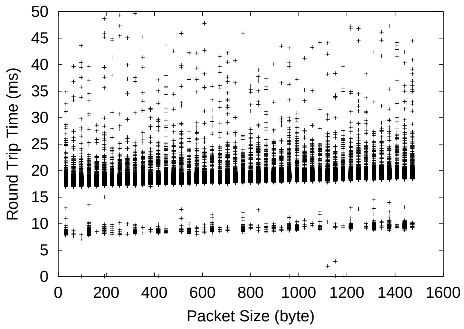

Figure 3: A sample having two groups of RTTs

proaches to obtain the accurate result. The approach to obtain the confidence interval is also described to increase the reliability on estimation. Those are presented in Subsection 3.1. Our clustering method to pick up proper RTTs from two or more groups of RTTs is then presented in Subsection 3.2. An adaptive mechanism to control the measurement period is finally presented in Subsection 3.3. Our experimental results based on those methods are shown in the next section.

3.1. Accurate and Reliable Slope Estimation Methods

As having been described in the previous section, using the linear least square fitting method in `pathchar` implies that errors follow a normal distribution. Thus, unexpectedly large errors (as shown in Figure 2) significantly affect the accuracy of the estimated value. To eliminate such an influence, we introduce two estimation methods instead of the linear least square fitting method. One is an M–estimation method with a Tukey's biweight function [9], which is a sort of the parametric approach. It is a robust estimation method to produce results with uniformly high efficiency. Because it presumes that almost all data is reliable and only some data includes unexpectedly large errors, the result is robust even if the large errors are contained as in our case. The other is a nonparametric linear least square fitting method which does not assume any distribution on measurement errors. In what follows, we will describe two methods in turn.

3.1.1. M–estimation Method

In this subsection, we describe the weighted least square fitting method. With this method, the influence of the large error can be limited. Note that this method is applicable when the number of large errors is rare but not negligible. Otherwise, we need to use a nonparametric approach which is independent of an error distribution. The latter approach is presented in the next subsection.

The M–estimation method is an extension of a maximum likelihood estimation method. In the M–estimation method, the weighted least square fitting is iterated to calculate an appropriate weight. There are some variations in the M–estimation method, and we apply the Turkey's biweight function which is considered to be one of the best estimation methods. In the Tukey's biweight function, a weight function is chosen as shown in Figure 4. It is apparent from the figure that the Tukey's biweight function is robust against the unexpectedly large errors if those occur infrequently. Let the number of kinds of the packet size be m. After we collect a minimum value of RTT for each packet size, we can estimate the slope according to the following procedure. Note that the slope means a coefficient β_n for router n (see Eq. (4)). In the following equations, we omit n for brevity.

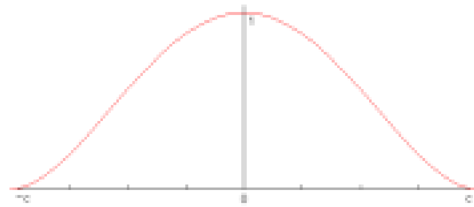
Figure 4: The biweight function

1. The straight line is expressed by $y = \alpha + \beta x$, where x is a packet size and y is an ideal minimum RTT. We set initial values of a vertical intercept α and a slope β with the least square fitting method;

$$\alpha = \bar{y} - \beta x, \quad \beta = \frac{\sum_{i=1}^{m} x_i y_i - m\bar{x}\bar{y}}{\sum_{i=1}^{m} {x_i}^2 - m\bar{x}^2}, \qquad (6)$$

   where $\bar{x}$ and $\bar{y}$ shows the mean of x_i and y_i.

2. Calculate the difference $|v_i|$ between the minimum RTT and the point on the straight line, i.e.,

$$|v_i| = |y_i - \bar{y}|. \qquad (7)$$

3. By obtaining the median of differences, the standard size of an error s is calculated as

$$s = median\{|v_i|\}. \qquad (8)$$

4. By using the biweight function, we set a weight adjustment factor ω_i^{adj} for each difference;

$${\omega_i}^{adj} = \begin{cases} \left[1 - (\frac{v_i}{c\,s})^2\right]^2 & \text{if } |v_i| < c\,s, \\ 0 & \text{otherwise} \end{cases} \qquad (9)$$

   where c is a constant value used as an index for making the total weight to be zero.

5. Let ω_i denote the weight of RTTs, which is defined by

$$\omega_i = \frac{m{\omega_i}^{adj}}{\sum_{i=1}^{m} {\omega_i}^{adj}}. \tag{10}$$

We then estimate new values of α and β with the weighted least square fitting.

$$\alpha = \frac{\sum_{i=1}^{m} \omega_i \, y_i}{m}, \quad \beta = \frac{\sum_{i=1}^{m} \omega_i \, x_i \, y_i}{\sum_{i=1}^{m} \omega_i \, x_i^2}. \tag{11}$$

6. After k iterations, we adopt α and β as solutions.

The parameter c in Eq. (9) controls a boundary for the errors contained in measured RTT values to be neglected. Through our experiments, we found that $c = 3$ and the number of iterations $k = 5$ are sufficient. Note that slopes of straight lines are always converged in our experimental results when we use above parameter values.

We introduce the following assumptions to calculate a confidence interval with the M-estimation method.

- For given packet size x, the random variable of the minimum RTT, Y follows the normal distribution, whose mean and variance are given by $\alpha + \beta x$ and σ^2, respectively.
- The number m of measurements for each packet size are mutually independent.

The above assumptions imply that the set of sloes follows the normal distribution with mean β and variance σ_B^2. We obtain those parameters by

$$\beta = \frac{\sum_{j=1}^{m} (x_j - \bar{x})(Y_j - \bar{Y})}{\sum_{j=1}^{m} (x_j - \bar{x})^2}, \tag{12}$$

$$\sigma_B^2 = \frac{\sigma^2}{\sum_{j=1}^{m} (x_j - \bar{x})^2}, \tag{13}$$

where σ^2 is the variance of the RTTs from the estimate line. It can be estimated from the measurement data as

$$\hat{\sigma}^2 = \frac{1}{m} \sum_{j=1}^{m} (y_j - \alpha - \beta x_j)^2. \tag{14}$$

From Eq. (12), we can calculate the values of the slope for links $n-1$ and n as $\hat{\beta}_{n-1}$ and $\hat{\beta}_n$, respectively. The variances for links $n-1$ and n are also estimated as $\hat{\sigma}_{n-1}^2$ and $\hat{\sigma}_n^2$ from Eq. (13).

Once those values are determined, we next calculate the confidence interval as follows. We can estimate the mean and variance for the difference of slopes as

$$\beta_u = \hat{\beta}_n - \hat{\beta}_{n-1}, \quad \sigma_u^2 = \hat{\sigma}_n^2 - \hat{\sigma}_{n-1}^2. \tag{15}$$

$1/\beta_u$ just gives a bandwidth estimation for link n. Results must follow t-distribution with $2m-4$ degrees of freedom. Thus, we first obtain the interval k as

$$k = \frac{c \, \sigma_u}{\sqrt{2m-4}}, \tag{16}$$

where c is a 97.5% value of the t–distribution if we want 95% confidence interval. Then we have the confidence interval for the estimated bandwidth $1/\beta_u$ as

$$1/(\beta_u + k) \le 1/\beta_u \le 1/(\beta_u - k). \tag{17}$$

We have a reliable estimation by adding the confidence intervals as described above. However, it is assumed that measurement errors follow the normal distribution after the very large errors are excluded by the biweight function. In the next subsection, we will present a nonparametric estimation method which does not require any assumption on the error distribution.

3.1.2. Nonparametric Estimation Method

In the nonparametric approach, we do not need any assumption on the error distribution. Let m be the number of obtained measurement data set for each packet size as before. The slope estimation can be obtained by the following procedure.

1. By choosing the every combination of two minimum values of RTTs, and calculate the slope. That is, we have $m(m-1)/2$ slopes by this step.
2. We sort a set of obtained slopes and adopt its median as the proper slope.

A Kendall's τ method [10] is known as a way of finding the confidence interval in the nonparametric method. However, it cannot be directly applied to the current problem since it is necessary to calculate the difference of two slopes. Alternatively, we use a Wilcoxon's method [11], which is based on the difference between medians of two data sets **S** and **T**.

In the current context, we use two sets of slopes obtained from the measurements for links $n-1$ and n, which are denoted as **S** and **T**, respectively. By letting the numbers of elements of **S** and **T** be $|S|$ and $|T|$, respectively, we label elements of two sets **S** and **T** as $s(j)$ and $t(i)$ $(1 \le i \le |T|, 1 \le j \le |S|)$. The bandwidth estimation and its confidence interval are then obtained as follows.

1. Calculate the set of differences $t(i) - s(j)$ $(1 \le i \le |T|, 1 \le j \le |S|)$. Let us denote the obtained set of the differences as **U**.
2. Sort the set **U**.

3. Let $u(i) (1 \leq i \leq |S| \times |T|)$ denote ith element of sorted set **U**. The confidence interval is then given by

$$u\left(\frac{|T|(2|S|+|T|+1)}{2}+1-a\right) \leq \beta_u \leq u\left(a-\frac{|T|(|T|+1)}{2}\right). \quad (18)$$

If we want 95% confidence interval, parameter a should be determined such that the probability $P(\sum u(i) \geq a)$ is equal to 0.975. When the numbers of measured data $|S|$ and $|T|$ are large, it is known that $\sum u(i)$ follows the normal distribution with mean $|T|(|S|+|T|+1)/2$ and variance $|S||T|(|S|+|T|+1)/12$. Thus, we can approximate a as;

$$a = \frac{|T|(|S|+|T|+1)}{2}+\frac{1}{2} + 1.96\sqrt{\frac{|S||T|(|S|+|T|+1)}{12}}. \quad (19)$$

We still have a problem in the above procedure. Our final goal is to control the measurement time so that the measurement is finished when the confidence interval of the bandwidth estimation is within a prespecified value. For that purpose, on–line calculation is necessary. However, the above procedure requires much computational time. Suppose that we gather RTT values with 45 kinds of packet sizes as in `pathchar`. The number of slopes obtained for each link becomes 990, and therefore the number of elements of **U** is about 1,000,000. It is too large for the method described above.

We therefore use another method based on a Kendall's rank correlation coefficient [10]. We obtain $m(m-1)/2$ slopes from m trials for each link, and therefore the number of elements $|S|$ and $|T|$ becomes $m(m-1)/2$. We therefore use the following procedure to estimate the confidence intervals.

1. Sort **S** and **T**, and obtain the set **U'**, the element of which is calculated by $u'(i) = s(i) - t(i)$ $(1 \leq i \leq m(m-1)/2)$.

2. The confidence interval of **U'** is then determined by the following equation.

$$u'\left(\frac{\frac{m(m-1)}{2}-C}{2}\right) \leq \beta_{u'} \leq u'\left(\frac{\frac{m(m-1)}{2}+C}{2}\right), \quad (20)$$

where C is the Kendall's rank correlation coefficient. If K is 97.5% value of the standard normal distribution and n is enough large, we obtain 95% confidence interval by using

$$C = K\sqrt{\frac{m(m-1)(2m-5)}{18}}. \quad (21)$$

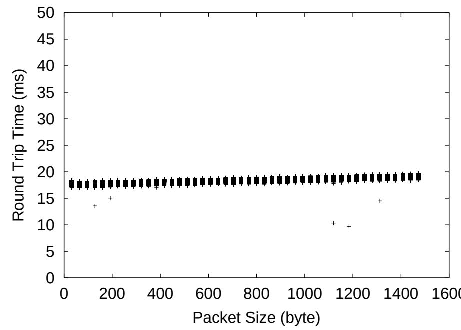

Figure 5: Result of clustering

The on–line calculation procedure and stopping rule for the RTT measurement will be described in Subsection 3.3.

3.2. Removal of Unnecessary RTT Values

As having been shown in Figure 3, it is necessary to pick up proper RTTs when the distribution of RTT consists of several groups of RTTs. It is caused by route alternation that `pathchar` can never detects. To divide data into several groups, we use the clustering method [12]. After we obtain the measurement data, we first abandon the upper 30% of measured RTTs since those does not help estimating the link bandwidth. Then, we divide them into several clusters.

We assume that the cluster having the largest number of measured elements contains the actual minimum RTT. If route alternation does not occur during the measurement, it is not necessary to apply the clustering. We can know it if divided clusters are very close with each other. Figure 5 plots the result of the clustering using the data shown in Figure 3. Note that we divided the gathered data into three clusters. The figure shows that we can extract the clusters of RTTs properly. A weak point of this procedure is that it takes much time for clustering and therefore we cannot repeat clustering for every packet arrival. From this reason, we perform clustering after the measurement of RTTs has been finished.

3.3. An Adaptive Mechanism to Control the Measurement Period

We finally describe our bandwidth estimation method. Our method can control the measurement period so that the measurement terminates when the prescribed confidence intervals are satisfied. During the measurement, inacurate data is dropped as described in the previous subsection. Then, one can rely on the obtained data with confidence intervals.

More specifically, the following procedure is performed during the RTT measurement. In describing the procedure below, we suppose that the bandwidth estimation for link $(n-1)$ has already been finished.

1. For estimating the bandwidth of link n, we first send a fixed number of packets. For example, we send 10 packets in our experiments presented in the next section. Then, RTTs are collected for 46 kinds of the packet size (from 40 bytes to 1,500 bytes). Namely, the source sends $10 \times 46 = 460$ packets for the initial measurement.

2. For taking account of route alternation, we check the source address of the ICMP packets as in `pathchar`. We take router n, the address of which appears most in the ICMP packets.

3. We then estimate the initial bandwidth and its confidence interval of n th link by either our estimation methods parametric or nonparametric method; (see Subsection 3.1).

4. To get the accurate bandwidth estimation and confidence interval, we iterate following procedures.

   (a) We send an additional set of probes (e.g., 10 packets for each packet size) to get new RTTs for router n.

   (b) For each RTT, we check whether the measured RTT is smaller than the minimum RTT. If so, we calculate the new values of the difference between measured RTT and the one derived from the estimated slope, and compare the new difference with the original one. If the difference is much smaller than the original difference, we replace the minimum value of RTT with the new RTT value. If the difference of new RTT is too large, it indicates that the route alternation occurs and the result doesn't make sense. In the experiment presented in the next section, we abandon the new RTT if the difference of it is larger than 30% of the difference of the previous RTT.

   (c) To keep the number of samples for each link to be identical, we send additional packets to router $(n-1)$ when the source sends more packets to router n.

   (d) By using our estimation approach (the parametric approach described in Subsection 3.1.1 or the nonparametric approach in Subsection 3.1.2), we update the bandwidth estimation and its confidential interval. The iteration terminates if the confidence interval of the estimated bandwidth becomes less than the prescribed value.

5. After the iteration terminates, we finally verify whether RTTs have reasonable values. A most important task at this step is to apply the clustering technique. If RTTs are not proper because of route alternation, we retry the measurement process by returning to Step 4.

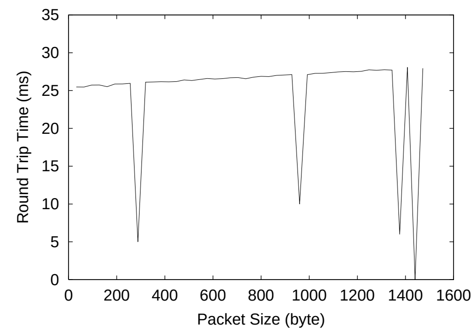

Figure 6: Minimum RTTs including irregular values

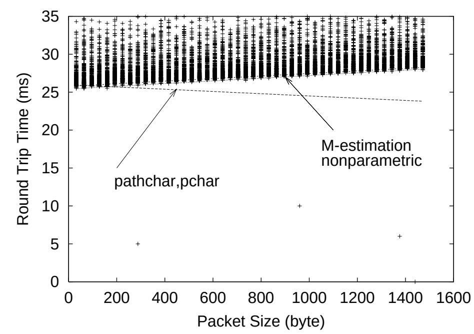

Figure 7: Estimated lines including irregular values

4. Experimental Results and Discussions

4.1. Removing Irregular RTT Values due to Exceptionally Large Errors

We first show experimental results for the case where RTT values apparently do not follow the normal distribution because of some large errors. The example was shown in Figure 2. Figure 6 plots only minimum values of RTTs against the packet size from Figure 2. As shown in this figure, the variation of minimum RTTs exhibits far from the linear relation. Figure 7 compares results of the slope estimations by `pathchar`, `pchar`, and our methods (the M-estimation and nonparametric methods). Straight lines of `pathchar` and `pchar` are inaccurate due to exceptionally large errors whose packet sizes are 288, 960, 1376, and 1440 bytes. On the other hand, our approach can filter out such errors.

Table 1 shows the estimated values. In the figure, two cases of the bandwidth estimation are shown; 13-th link from 202.231.198.2 destined for 210.142.124.1 (corresponding to Figure 2) and 13-th link from 202.232.8.66 destined for 210.141.224.162. The capacities of those links

Table 1: Bandwidth estimation and confidence intervals for the measurement data with irregular values

BW	method	estimated results	sent
1.5M	Pathchar	0.87	200
	M-estimation	$1.34 \leq 1.35 \leq 1.36$	10
	Wilcoxon	$1.32 \leq 1.33 \leq 1.36$	20
	Kendall	$1.30 \leq 1.33 \leq 1.37$	20
45M	Pathchar	86.65	200
	M-estimation	$44.06 \leq 46.58 \leq 49.42$	200
	Wilcoxon	$42.99 \leq 53.44 \leq 66.54$	200
	Kendall	$52.22 \leq 53.44 \leq 54.69$	200

were known a priori as 1.5 Mbps and 45 Mbps. For each of two links, we show the estimated bandwidth obtained by all methods. Confidence intervals of 95% are also shown in our methods. As shown in the table, results obtained by `pathchar` and `pchar` are far from the actual bandwidth, while our methods can give very close values. The difference of the actual bandwidth and the estimated bandwidth is due to the overhead of the underlying network. In the case of 45 Mbps link, our methods seem to offer very accurate estimation. Perhaps, it is not true since we must exclude the overhead of the underlying network.

In the table, the numbers of packets transmitted for each packet size are also shown. In our methods, the very small number of packets were sufficient to obtain the accurate results for 1.5 Mbps link. For 45 Mbps link, on the other hand, 200 packets were necessary, which is same as `pathchar`. It is due to the fact that as the link bandwidth becomes large, the accurate estimation becomes difficult, which has already been pointed out in [4].

In the case of 1.5 Mbps link, we cannot observe differences among our three methods, the M-estimation, Wilcoxon's and Kendall's methods. In the case of 45 Mbps link, the M-estimation method seems to be best. However we cannot decide the best one here because we found many cases that the other method gives the best result, as will be presented in the below.

4.2. Clustering RTT Values against Route Alternation

If the distribution of RTT values consists of several groups due to route alternation, it is apparent that the approach to cut off the exceptionally large errors mentioned above is not sufficient. See Figure 8, where we plot minimum values of RTTs against the packet size. The RTT values fluctuate to a large extent. Of course, it misleads us about the estimation, and the estimation obtained by `pathchar` and `pchar` are meaningless. Then, our clustering approach presented in

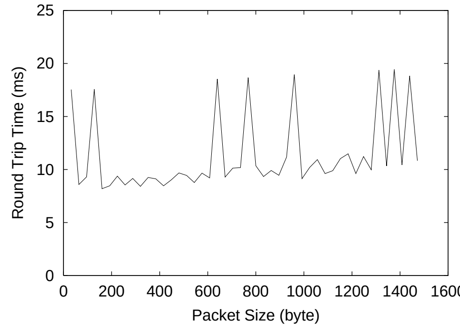

Figure 8: Minimum RTT for mixed groups

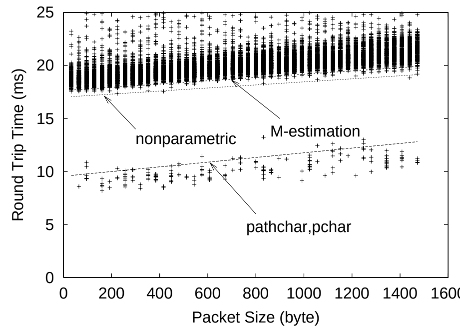

Figure 9: Estimated slopes in the case of mixed RTT values

Subsection 3.2 becomes necessary to exclude the RTT values obtained by an *exceptional* route.

Table 2 shows the estimation results. In the table, the cases of 10 Mbps and 12 Mbps links are shown. Those are located at 8-th link from 150.100.59.2 towards 202.219.160.22 and 15-th link from 210.157.131.158 towards 210.224.236.1. The numbers of packets transmitted in each method are also shown in the table. From this table, our estimations show the reasonable values while `pathchar` and `pchar` lead to even a negative value. The reason becomes clear when we look at the slope estimation plotted in Figure 9. Since the estimated values of `pathchar` and `pchar` is small, the resultant estimation on the bandwidth of the target link takes a negative value. On the other hand, our methods can estimate the slope adequately.

However, estimation results obtained by our methods are not satisfactory as shown in Table 3. For fair comparison, we set the number of transmitted packets to be 200 in all cases. By clustering the data set in our method, several measurement data were excluded. Then, the used measurement data was not sufficient to obtain the reliable result. In our examination, about 7% of collected data (628 packets out of $200 \times 46 = 9200$ packets) was unused. Since our current clustering method is computationally intensive, online calculation is impossible to adaptively increase the number of

Table 2: Bandwidth estimation and confidence intervals for the case of mixed RTTs

BW	method	estimated results	sent
10M	Pathchar	-22.6	200
	M–estimation	$10.07 \leq 12.40 \leq 16.11$	200
	Wilcoxon	$16.59 \leq 16.95 \leq 24.07$	200
	Kendall	$14.24 \leq 16.95 \leq 25.29$	200
12M	Pathchar	8.25	200
	M–estimation	$9.79 \leq 9.94 \leq 10.09$	20
	Wilcoxon	$13.3 \leq 13.8 \leq 14.4$	90
	Kendall	$13.6 \leq 13.8 \leq 14.1$	90

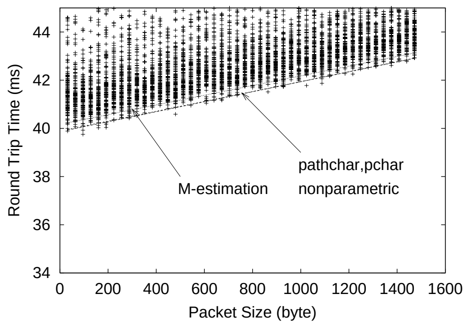

Figure 10: RTT values and estimated slopes

Table 3: Bandwidth estimations and confidence intervals

BW	method	estimated results	sent
6M	Pathchar	5.75	200
	M–estimation	$6.48 \leq 6.60 \leq 6.72$	200
	Wilcoxon	$5.65 \leq 5.87 \leq 5.92$	200
	Kendall	$5.67 \leq 5.87 \leq 5.94$	200
1.5M	Pathchar	1.46	200
	M–estimation	$1.37 \leq 1.40 \leq 1.43$	20
	Wilcoxon	$1.43 \leq 1.45 \leq 1.47$	110
	Kendall	$1.42 \leq 1.45 \leq 1.48$	110
12M	Pathchar	10.6	200
	M–estimation	$- \leq 10.5 \leq -$	200
	Wilcoxon	$10.40 \leq 11.34 \leq 12.28$	50
	Kendall	$11.04 \leq 11.34 \leq 11.59$	50

Table 4: Variations on the required number of probes

bandwidth	link	M–estimation	Nonparametric
10 Mbps	10 th	10	630
10 Mbps	12 th	*1127	220
12 Mbps	15 th	10	10
12 Mbps	12 th	30	80
45 Mbps	13 th	370	*1007
100 Mbps	16 th	427	979
100 Mbps	9 th	*1080	*1080

sample data. We need more research on this aspect.

4.3. Controlling the Measurement Period Adaptively

We next show how our adaptive control of the measurement period works. Different from previous cases, we pick up the cases where `pathchar` can also show the reasonable results in this subsection. Figure 10 compares estimated slopes of minimum RTTs among `pathchar`, `pchar` and our methods. As shown in the figure, there is no remarkable difference among all estimation methods. Table 3 also shows the same tendency; estimated values of link bandwidths are quite close with each other. These results suggest that the error contained in the minimum values of RTT can be modeled by a normal distribution in usual cases if the amount of measurement data is sufficiently large.

However, our estimation approaches have two advantages over `pathchar` (and `pchar`). First, our method can control the number of probes adaptively. As shown in Table 3, the measurement terminates with a less number of probes in our method except the case of 6 Mbps link. Table 4 summarizes the required number of probes to obtain the 95% confidence intervals where minimum and maximum values are within 5% difference from the mean value. Note that symbol '*' in the table shows that the result does not reach within the prescribed confidence interval by that number of probes. For several links, the number of probes for each packet size is less than 200. On the other hand, the number of transmitted probes by `pathchar` was always 200; it implies that `pathchar` wastes the network bandwidth by unnecessarily transmitting packets. For other cases, the numbers of probes are larger than `pathchar`, but we can expect that the resultant estimated values become more reliable than the values obtained by `pathchar`.

A second advantage of our methods is that we can obtain unified degrees of confidence on all links. On the other hand, the accuracy of estimation by `pathchar` is varied, and more importantly, there is no means to know about reliability on the estimated values.

Between parametric and nonparametric approaches, the required number of probes by the nonparametric approach is larger than that of the parametric approach. It is natural since the nonparametric approach does not assume any distribution on errors. Then, it needs a larger number of probes for reliable estimation. The large number of probes was necessary for the second link in the table in spite of 10 Mbps link. It is because the utilization of that link was high. It verifies that our method can adaptively increase the number of probes according to the link congestion.

4.4. Online Estimation of Confidence Intervals

We last discuss on the derivation methods of confidence intervals in our methods. As having been described in Section 3.1.2, the method based on Kendall's rank correlation coefficient is approximate in obtaining the confidence interval, and it must be less accurate than the one based on Wilcoxon's method. However, differences between Kendall's and Wilcoxon's methods were within 5% of the link bandwidth as having been shown in Tables 1, 2, and 3. If we collect p kinds of the packet size, calculation time by Wilcoxon's method becomes $O(p^4)$, while $O(p^2)$ in Kendall's method. Therefore, Kendall's method is useful for the online estimation of confidence intervals.

In our experiments, the M–estimation method sometimes failed to determine the confidence interval, which was shown in the last example of Table 3. It is caused by assuming that the variance of slopes σ_n^2 for link n is larger than σ_{n-1}^2 for link $(n-1)$. See Eq. (15). That assumption is valid if we can measure RTTs of routers $n-1$ and n by the same packet. However, because it is impossible, RTTs of routers $n-1$ and n must be measured separately, and the above assumption does not hold.

As having been presented in the tables, the assumption that the measurement errors follow the normal distribution seems to be often valid. However, it can only be known by the links, bandwidth of which is a priori known. Therefore, we should use the nonparametric approach to obtain a reliable estimation.

5. Conclusion

We have explained the bandwidth estimation method based on `pathchar` and more recent `pchar`, and proposed two bandwidth estimation methods. From experimental results, we have shown that our methods can produce the robust estimations. Our findings are as follows;

1. `Pathchar` cannot estimate the bandwidth adequately due to two kinds of unexpected errors; a few but very large errors and route alternation. Those pose that measurement errors do not follow some probability distribution such as a normal distribution.

2. We can eliminate exceptionally large errors by utilizing the biweight estimation method, which is applicable to both of M–estimation and nonparametric least square fitting methods.

3. By clustering the measured RTTs and selecting an appropriate cluster, errors introduced by route alternation can be avoided.

4. By obtaining the confidence interval, a measurement period can be controlled, which makes it possible to reduce the measurement period and avoid bandwidth waste caused by unnecessary probes in some cases. If the link is congested, on the other hand, more probes are transmitted according to our method. Then accurate and, more importantly, reliable estimation becomes possible.

5. Between parametric and nonparametric approaches, the latter is adequate for reliable bandwidth estimation, but it requires more measurement time. The parametric approach (i.e., the M-estimation method) is better in the measurement and computational time. Perhaps, it depends on the link condition. If the link load is not high, the obtained measurement data is stable. Then, the assumption that the measurement errors follow the normal distribution would hold. Otherwise, the nonparametric approach presented in this paper would be necessary. However, its validation remains as a future research topic.

References

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wangm, and W. Weiss, "An architecture for differentiated services," *IETF RFC 2475*, December 1998.

[2] E.C.Rosen, A.V.Than, and R.Callon, "Multiprotocol label switching architecture," *IETF draft*, `http://search.ietf.org/internet-drafts/draft-ietf-mpls-arch-06.txt`, March 1998.

[3] "Caida measurement tool taxonomy," `http://www.caida.org/Tools/taxonomy.html`

[4] A. B. Downey, "Using pathchar to estimate Internet link characteristics," *Proceedings of ACM SIGCOMM*, pp. 241–250, August 1999.

[5] V. Jacobson, "pathchar," `ftp://ftp.ee.lbl.gov/pathchar/`.

[6] V. Jacobson, "traceroute," `ftp://ftp.ee.lbl.gov/traceroute.tar.Z`

[7] B. A. Mah, "pchar: A tool for measuring Internet path characteristics," `http://www.ca.sandia.gov/~bmah/Software/pchar`.

[8] V. Paxon, "Measurements and analysis of end-to-end Internet dynamics," *Ph.D.Thesis, University of California Berkley*, April 1997.

[9] J.W.Tukey, "Introduction to today's data analysis," *Proceedings of the Conference on Critical Evaluation of Chemical and Physical Structural Information*, pp. 3–14, 1974.

[10] M.G.Kendall, *Rank Corelation Methods*. Griffin, 1970.

[11] G.Noether, "Some simple distribution–free confidencd intervals for the center of a symmetric distribution," *J. Am. Statist. Assoc.*, pp. 716–719, 1973.

[12] J. A. Hartigan, *Clustering algorithms*. John Wiley & Sons, 1975.