J Netw Syst Manage (2013) 21:623–649
DOI 10.1007/s10922-013-9266-4

On the Accuracy of Packet Delay Estimation in Distributed Service Networks

Nafei Zhu · Jingsha He · Yue Zhou · Wei Wang

Received: 29 January 2012 / Revised: 27 December 2012 / Accepted: 5 February 2013 /
Published online: 14 February 2013

Abstract Packet delay (either one-way time or round-trip time) is a very important metric for measuring the performance of networks in a highly dynamic environment such as the Internet. Many network applications are also sensitive to packet delay or delay variation for ensuring an acceptable level of quality in providing network-based services such as VoIP, multimedia streaming, etc. A very important property of packet delay is that it is very dynamic and therefore should be measured frequently with measurement results being updated on a timely basis. Measurement of packet delay has thus generated a great deal of interest in the past years and a lot of research has been performed in the development of measurement architecture as well as specific measurement techniques. However, how to reduce network overhead resulting from measurement while achieving a reasonable level of accuracy still remains a challenge. In this paper, we propose to use delay estimation as an alternative to delay measurement for reducing measurement overhead and, in particular, examine the level of accuracy that delay estimation can achieve. With delay estimation, measurement nodes can be dynamically selected and activated and other nodes can share measurement results by performing delay estimation, thus reducing measurement overhead while supporting the dynamic requirement for delay measurement. Consequently, while measurement overhead can be reduced by activating

N. Zhu · J. He (✉)
Beijing University of Technology, 100 Ping Le Yuan, Chao Yang District, Beijing 100124, China
e-mail: jhe@bjut.edu.cn

N. Zhu
e-mail: zyy@emails.bjut.edu.cn

N. Zhu · Y. Zhou · W. Wang
National Application Software Testing Labs, Zhongguancun Software Park, Building 3A, Shangdi, Haidian District, Beijing 100193, China
e-mail: zhouyue@bsw.net.cn

W. Wang
e-mail: wangwei@bsw.net.cn

only a subset of network nodes to perform actual measurement, desired accuracy can be achieved by exploring the correlation between delays as well as by sharing measurement results to do delay estimation based on such a correlation. We illustrate how packet delays of network nodes can correlate to each other based on topological properties and show how delays can be estimated based on such a correlation to meet accuracy requirements, which would make delay measurement in the Internet highly dynamic and adaptable to the accuracy requirements and measurement results highly reliable. We also show how delay estimation can be applied by presenting three application scenarios as well as an example to demonstrate the usefulness and effectiveness of delay estimation in the measurement of packet delays.

Keywords Network management · Measurement · Round trip time · Correlation · Internet

1 Introduction

Packet delay between network nodes in the Internet is a very important metric in network management, for it can be used to characterize network performance and has thus become an important part of network measurement. There has been some previous work in this area. IDMaps is probably the first tool developed for estimating the latency of arbitrary network paths [1]. Gummadi et al. [2] proposed a tool called King to estimate the latency between arbitrary end hosts by measuring the latency between nearby DNS name servers. Turbo King (T-King) was proposed to improve King by streamlining the process of making distance measurements using DNS, resulting in improvement of accuracy and reduction of overhead [3]. Hariri et al. [4] proposed a decentralized coordinate-based solution for Internet delay measurement by modeling the latency between each pair of nodes as a virtual distance between these nodes in a virtual coordinate system. Sharma [5] presented a network proximity and latency estimation tool for locating the closest node to a given node. Agarwal and Lorch [6] designed a latency prediction system for game matchmaking scenarios. Xing and Chen [7] proposed a virtual node based network distance prediction mechanism in which edge networks are divided into independent prediction regions and each prediction region has a virtual node to represent the joint between the region and the core network. The distance of any two nodes is then represented as the sum of the distances of the two nodes to their representative virtual nodes and that between the two virtual nodes. Telemeter is a network prediction system based on network topology [8] that can predict the distance between any pair of IPv6 hosts that connect to the IPv6 network through CERNET2 using inter-domain/intra-domain routing information provided by ISPs. “NetForecast” is an algorithm for delay prediction in provider controlled networks [9] that is based on a combination of landmark-based distance estimation, clustering and a triangulation principle. There is also a lot of other work in this area, such as ICS [10], Virtual Landmark [11], and so on.

Some of the early work has relied on the notion of network coordinates in which an effort is made to embed the Internet into a Euclidean space. Then, each network node is assigned a coordinate based on its topological position and the distance between any two nodes can then be computed using the Euclidean distance function. Although a number of network coordinate systems have been proposed, it has been shown that the accuracy of distance prediction is far from being satisfactory [12]. Some of the other work on delay measurement and estimation is based on the technique of clustering network nodes together. Then, a core node in each cluster is selected as the measurement node and other nodes in the cluster simply take the delay measured by the core node as their estimated delay value. Thus, delay estimation using such methods cannot provide the desired granularity of prediction accuracy. Neither can they support the dynamic selection of measurement nodes to achieve the desired accuracy. Meanwhile, we have noticed that delays for different network nodes may exhibit some correlation to each other due to their inherent topological properties. Such correlations can allow us to design an Internet measurement architecture to help reduce measurement overhead while deriving quantitative delay results with desired accuracy by exploring such topological properties. It can also allow us to decide where to deploy measurement nodes in an optimal way and how to dynamically activate measurement nodes based on accuracy requirements from various applications.

Although Iso-bar discussed correlations between landmarks [13], the work is not based on exploring the underlying network topology. So far, we have found little work in delay measurement in which topological properties of network nodes are considered for accurate delay measurement and estimation. It is, therefore, the purpose of this paper to study how accurate delay estimation can achieve compared to delay measurement and how delay estimation techniques can be actually applied in various network scenarios. The result of our study would then allow us to develop a measurement architecture in which measurement nodes can be dynamically selected to perform an actual delay measurement while other "neighboring" nodes can share the measurement results and conduct a delay estimation based on their topological relationships with the measurement nodes to meet the dynamic requirements for delay measurement, i.e., frequent measurement, low overhead, and desired level of accuracy for the results.

In this paper, we study the correlation of packet delays between network nodes based on their topological relationship [14]. We carry out the study based on experiment data from using OPNET [15]. In particular, we study the relationship between two sequences of delay values from two source nodes to a common destination node and derive a correlation coefficient that reflects the relationship. Then, based on such a delay correlation, we derive a formula to express the accuracy of delay estimation by relating accuracy with the correlation coefficient. Third, we calculate accuracy and deduce some main characteristics for delay estimation. Fourth, we find the relationship between accuracy and path length ratios that can help further reduce the work on delay estimation. Finally, we describe three algorithms that deal with three different problems in delay estimation to illustrate how to apply delay estimation. We also use examples to demonstrate the usefulness and effectiveness of our algorithms.

The rest of this paper is organized as follows. In the next section, we state our research goal and explain our experiment methodology. In Section 3, we introduce the formula that characterizes correlation as well as some notations. In Section 4, we study the relationship between the correlation of delays and accuracy for delay estimation and determine the method for calculating accuracy. In Section 5, we analyze and derive several characteristics about accuracy for delay estimation by exploring the relationship between accuracy and the Kendall coefficient. In Section 6, we evaluate our methodology using real network data to show the effectiveness of delay estimation. We also compare our estimation model with other alternative methods and show the advantages of our method over those methods. In Section 7, we describe a framework for the measurement and estimation of network delays, present three application scenarios for the delay estimation problem, and illustrate the corresponding accuracy results using an example. Finally, in Section 8, we conclude this paper in which we also discuss our future research.

2 Problem Statement and Experimental Strategy

2.1 Problem Statement

We define the correlation of packet delays between two source nodes to a common destination node as a metric that describes to what extent the two sequences of delay values exhibit the same changing trends. In this paper, we focus on studying the accuracy for delay estimation based on such a correlation and apply the result to the estimation of packet delays based on the accuracy requirement and measurement values of packet delays.

2.2 Experiment Methodology

We decide to use PING to get delay values. Consequently, all the delay values are round-trip times (RTTs). In the experiment, we perform simulation with OPNET [15] to get RTT values with defined hops and network loads. A typical simulation topology is shown in Fig. 1 in which there are four subnets: two of which contain the two respective source nodes that send PING packets and the third contains the common destination node for the PING packets. The subnet in between contains the node that merges the two paths from the two source nodes into one common path. In every subnet, the workstation sends or receives PING packets and the Raw Packet Generator (RPG) module in the OPNET generates self-similar traffic as the background traffic. Of course, there may be one or more other subnets between the source and the destination nodes as long as the basic topological properties of the nodes and paths are maintained.

2.3 Parameter Configuration

There are at least three parameters that should be configured for the experiment.

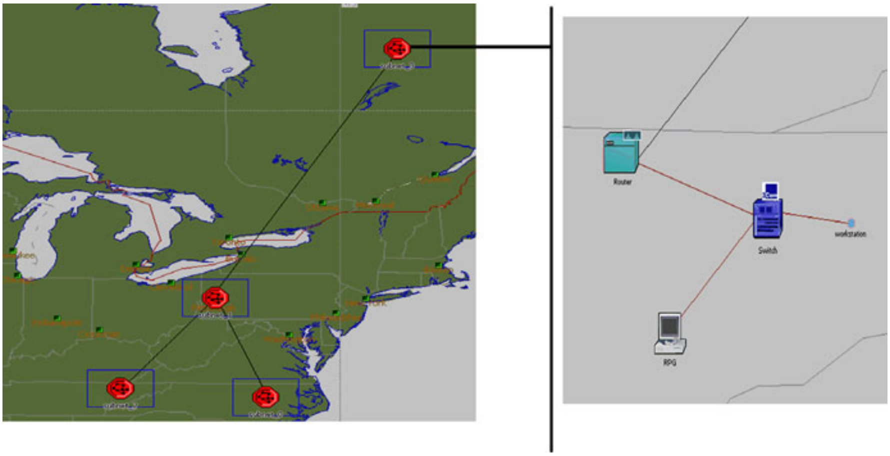


Fig. 1 The simulation topology

2.3.1 Path Length

The average number of hops in the Internet is 15–19 according to Fei et al. [16] and the default maximum path length is 15 hops in OPNET. So, in our experiment, we set the maximum path length for PING to be 15 hops to make our experiment as more representative as possible.

2.3.2 Path Load

We model the load on the paths in three representative levels: light, medium and heavy with an average utility of the path being 20, 50 and 80 %, respectively, and within the range [15–25 %], [40–60 %] and [65–95 %], respectively. The purpose of setting the traffic load at several different representative levels is to study how traffic load can affect the accuracy of delay estimation. This is important because it helps us better understand how delay estimation performs in comparison to delay measurement in various network traffic situations in general and how our proposed delay estimation method would respond to different network traffic situations in particular.

2.3.3 Traffic Pattern

The traffic pattern in the Internet has been shown to be self-similar rather than Poisson [17]. We therefore use the RPG module in the OPNET to generate self-similar traffic with a typical Hurst value of about 0.7 [18].

We run each experiment for 30 min, which is the default for many experiments [18].

2.4 Experiment Data

Based on the above methodology and configuration, we performed 11,259 experiments with OPNET to cover all possibilities of path loads and path lengths and, as a result, obtained 2,748,816 data based on which we proceeded to perform our study and propose our solutions.

3 Basic Formula and Notations

3.1 Formula for Correlation

We use rank-based Kendall's method to define and calculate correlation τ [19] which is defined as follows:

$$\tau = 1 - \frac{2s(\pi, \sigma)}{N(N-1)/2} \tag{1}$$

According to Kendall's definition, if $Y = \{y_1, y_2, \ldots, y_N\}$ is a set of ordered items and π and σ denote two distinct orderings of Y, $s(\pi, \sigma)$ is then the minimum number of adjacent transpositions needed to bring π to σ. It has been shown that Kendall's method is less sensitive to outliers and non-normality than the standard Pearson estimate [20].

3.2 Path Length and Path Load

Let A and B be the two source nodes, T be the common destination node, and M be the node that joins paths AT and BT. Consequently, paths AT and BT share a common path TM whose length is H_{com} and the load on the common path is U_{com}. Similarly, the length of the private path AM is H_{pri_a} and the load on the path is U_{pri_a}. The same can be defined for the private path BM. In this paper, $H_{com} + \max(H_{pri\_a}, H_{pri\_b}) \le 15$ and $0 \le U_{com}, U_{pri\_a}, U_{pri\_b} \le 100\ \%$. We call "$H_{com}$, H_{pri_a}, H_{pri_b}" a length combination (LEC for short) and "U_{com}–U_{pri_a}–U_{pri_b} %" a load combination (LOC for short).

For the path length, we always assume that $H_{com} \ge 1$ and $\min(H_{pri\_a}, H_{pri\_b}) \ge 1$. This is because when $H_{com} = 0$, it is hard to measure the correlation between the delays over AT and BT due to the lack of any commonality between the two paths. Consequently, the total number of LECs in our experiment is 417.

For the path load, we have selected three typical values in our study as explained in Sect. 2. In total, there are 27 LOCs. Moreover, in this paper, we assume general network load for simplicity of analysis. Therefore, only three LOCs remain, i.e., 20–20–20 %, 50–50–50 % and 80–80–80 %.

4 Accuracy of Delay Estimation

We have shown that the correlation of packet delays for AT and BT is dominated by LECs and can be described using the length ratio, which is defined as the ratio of the common path length to the longer private path length [14] under any given network load. According to the above property, we would like to utilize the correlation and find a method for estimating one delay based on another that is presumably a measured delay value. The length ratio property indicates that, in general, the higher the length ratio is, the bigger the correlation coefficient gets, which expresses the relationship between the two delay values. Furthermore, the bigger the correlation coefficient is, the closer the relationship gets and, therefore, the more accurate the delay estimation can achieve. This conforms to the essence of the correlation coefficient that expresses the extent to which two variables are linearly related to each other. By applying the significant test for regression equation [21], our experiment data show that all the p values fall below 0.05 when the length ratio becomes higher than 3, thus allowing us to model the relationship between the two sequences of RTT values in a linear manner. Based on such a relationship, we can develop a method to estimate delay from one source node to a destination node based on a measured delay value from another source node to the same common destination node. And the method can be quite comprehensive and representative.

4.1 Analysis of Accuracy Formulas

We intend to derive a liner expression to express the relationship between two RTT sequences (x_1, x_2, …, x_n) and (y_1, y_2, …, y_n). With the measured values (x_1, y_1), (x_2, y_2), …, (x_n, y_n), the regression formula can be expressed as follows:

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i + \varepsilon_i$$

where

$$E(\varepsilon_i) = 0,\ \operatorname{var}(\varepsilon_i) = \sigma^2,\ i = 1, 2, \cdots n \tag{2}$$

We define accuracy for the estimation of delay y_i in the form of a relative error:

$$re_i = \left|\frac{y_i - \hat{y}_i}{y_i}\right| \text{ and } \overline{re_i} = \overline{\left|\frac{y_i - \hat{y}_i}{y_i}\right|} \tag{3}$$

in which y_i is the measured value and $\hat{y}_i$ is the estimated value. Accuracy is just the complement of the relative error, i.e., $1–re_i$.

It is known that the square of Pearson coefficient R, which is also called the coefficient of determination, is often used in statistical methods to explain how much of the variability of a factor can be caused or explained by its relationship to another factor [21]. That is:

$$R^2 = \frac{\sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \text{ and } \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

Then,

$$\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = (1 - R^2) \sum_{i=1}^{n} (y_i - \bar{y})^2$$

and

$$\overline{(y_i - \hat{y}_i)^2} = \frac{1}{n}(1 - R^2)\sum_{i=1}^{n}(y_i - \bar{y})^2 \text{ for } \varepsilon_i \sim N(0, \sigma^2) \text{ so } E(\varepsilon_i) = 0.$$

Therefore,

$$\overline{(y_i - \hat{y})}^2 = \overline{(y_i - \hat{y})^2} = \frac{1}{n}(1 - R^2)\sum_{i=1}^{n}(y_i - \bar{y})^2$$

and

$$\overline{\left|\frac{y_i - \hat{y}_i}{y_i}\right|} \approx \overline{sqrt\left(\frac{1}{n}(1 - R^2)\sum_{i=1}^{n}(y_i - \bar{y})^2\right)/|y_i|} \tag{4}$$

Using our experiment data, we explored the relationship between Eqs. (3) and (4) and the result is shown in Fig. 2 in which the x axis shows the values calculated using Eq. (3) and the y axis shows the values using Eq. (4). We can thus see that they exhibit a linear relationship. We can also perform a linear regression analysis for these two sets of values and, according to [21], calculate the Multiple R-Squared to be 0.8471 and the Adjusted R-squared to be 0.8468, both of which are high values. So, we can conclude that they exhibit a liner relationship. Moreover, since the values calculated using (3) are smaller than those using (4), we can use Eq. (4) to substitute Eq. (3) when calculating the relative error; for using Eq. (4) doesn't change the main characteristics of Eq. (3) but can provide a more conservative estimation on accuracy.

4.2 The Accuracy Formula to be Used

Although we can use Eq. (4) to substitute Eq. (3), the R in Eq. (3) is a Pearson coefficient rather than a Kendall coefficient which we used in our earlier work [13] to characterize correlation. Therefore, we need to find a relationship between these

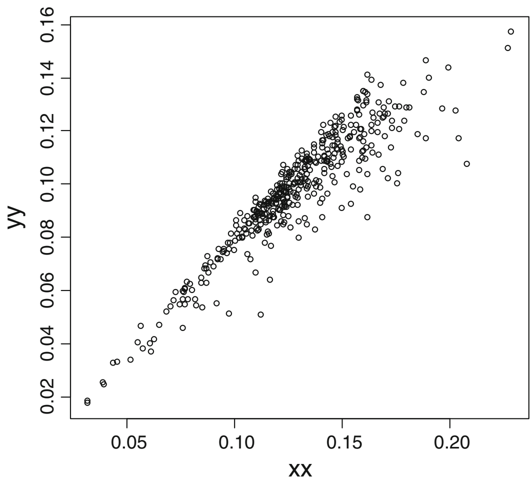


Fig. 2 Results from using the two equations for calculating relative errors

two coefficients. For such a purpose, we calculate the Pearson and the Kendall coefficients based on all the RTT values in our experiment and show the relationship between them in Fig. 3. From the figure, we can see that the relationship can be modeled in a linear manner. We thus apply the linear regression and the resulting line is also shown in the figure.

Although the linear regression line shows some deviations when the Kendall coefficient is bigger than 0.7, almost all the measured values fall under the regression line. Therefore, it won't make the relative error of estimation any worse. Thus, we can model the relationship between these two coefficients as a linear one. From the regression formula, we can see that the Kendall coefficient is smaller than the Pearson coefficient. Consequently, using the former to substitute the latter won't compromise the accuracy of delay estimation. As the result, we will use the Kendall coefficient as the R in Eq. (4), that is,

$$1 - \overline{\left|\frac{y_i - \hat{y}_i}{y_i}\right|} \approx 1 - \overline{sqrt\left(\frac{1}{n}(1 - T^2)\sum_{i=1}^{n}(y_i - \overline{y})^2\right)/|y_i|} \tag{5}$$

5 Characteristics of Accuracy for Delay Estimation

We now study some main characteristics that can help us determine accuracy for delay estimation. We should notice that there are always two paths in each analysis that result from two source nodes to the same (common) destination node. If the lengths of the two paths are not the same, we can examine the accuracy by using the delay over the longer path to estimate that over the shorter path, and vice versa.

5.1 Longer Path Characteristic

We calculate the relative error for each LEC under any of the three LOCs using Eq. (5). In our delay correlation analysis, there are always three paths in an LEC and

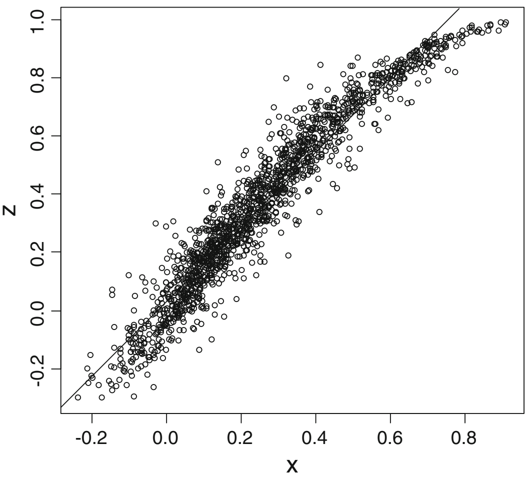


Fig. 3 Relationship between Pearson and Kendall coefficients

changing the length of any will make it a new LEC. Even after we fix the lengths of the common and the longer private paths, there will still be one or more LECs resulting from varying the length of the shorter private path. In total, there are 763 $\times$ 3 accuracies to analyze. We find that for LECs that have the same common and longer private path lengths, the accuracies are almost the same. The results are shown in Fig. 4 in which (a) is the case for estimating the delay of the shorter path based on the delay value of the longer path and (b) is that for estimating the delay of the longer path based on the delay value of the shorter path.

We can see from Fig. 4 that accuracy improves as the common path length gets longer, which verifies that accuracy has an inverse mutation with a Kendall coefficient. At the same time, accuracies are not much different for the LECs in which the common and the longer private path lengths remain unchanged.

We would now use the variance analysis to support the results in Fig. 4 in which we show that different shorter private path lengths won't cause significant difference to accuracy. For variance analysis, we arrange the accuracies in a three-dimensional matrix in which the first dimension contains the 7 different private path lengths denoted as a:a, a:a–1, …, a:a–6 as shown in Fig. 4, the second dimension contains the 14 different values for the common path length due to our assumption that the sum of the common path length and private path length cannot be longer than 15, and the third dimension contains the 3 LOCs. Then, we extract 14 two-dimensional arrays from this three-dimensional matrix. In each array, the rows are the 3 LOCs and the columns are the number of private path lengths that have non-zero values for the accuracy. We then perform the variance analysis for each of the 14 arrays. When the common path length varies from 1 to 12, the corresponding arrays will have 3 or more columns, allowing us to do the analysis with the Friedman test [21]. Let R_{ij} be the rank of x_{ij} within block i. The test statistic for the Friedman test can be carried out using the following formulas:

$$Q = \frac{12N}{s(s+1)} \sum_{i=1}^{s} \left(R_{i\cdot} - \frac{1}{2}(s+1) \right)^2 \tag{6}$$

$$R_{i\cdot} = \frac{1}{N}(R_{i1} + R_{i2} + \cdots + R_{iN}),\ i = 1, 2, \cdots, s \tag{7}$$

When the common path length is 13, the corresponding arrays only has 2 columns, which allows us to do the analysis with the Binomial test [21] that is more suitable for verifying whether two sets are significantly different. Let p be the probability for getting one category. Then q = 1–p is the probability for getting the other category. The formula for the Binomial test is as follows:

$$p(r)_{binomial} = {}_nC_r * p^r * q^{n-r} = \frac{(n!p^r q^{n-r})}{(r!(n-r)!)} \tag{8}$$

We don't perform any analysis when the common path length is 14 since the corresponding array will then only have 1 column. The threshold value for the variance analysis for both scenarios is 0.05 [21] and the results are shown in Fig. 5 in which (a) is the case for estimating the delay of the shorter path based on the delay of the longer path while (b) is the reverse case.

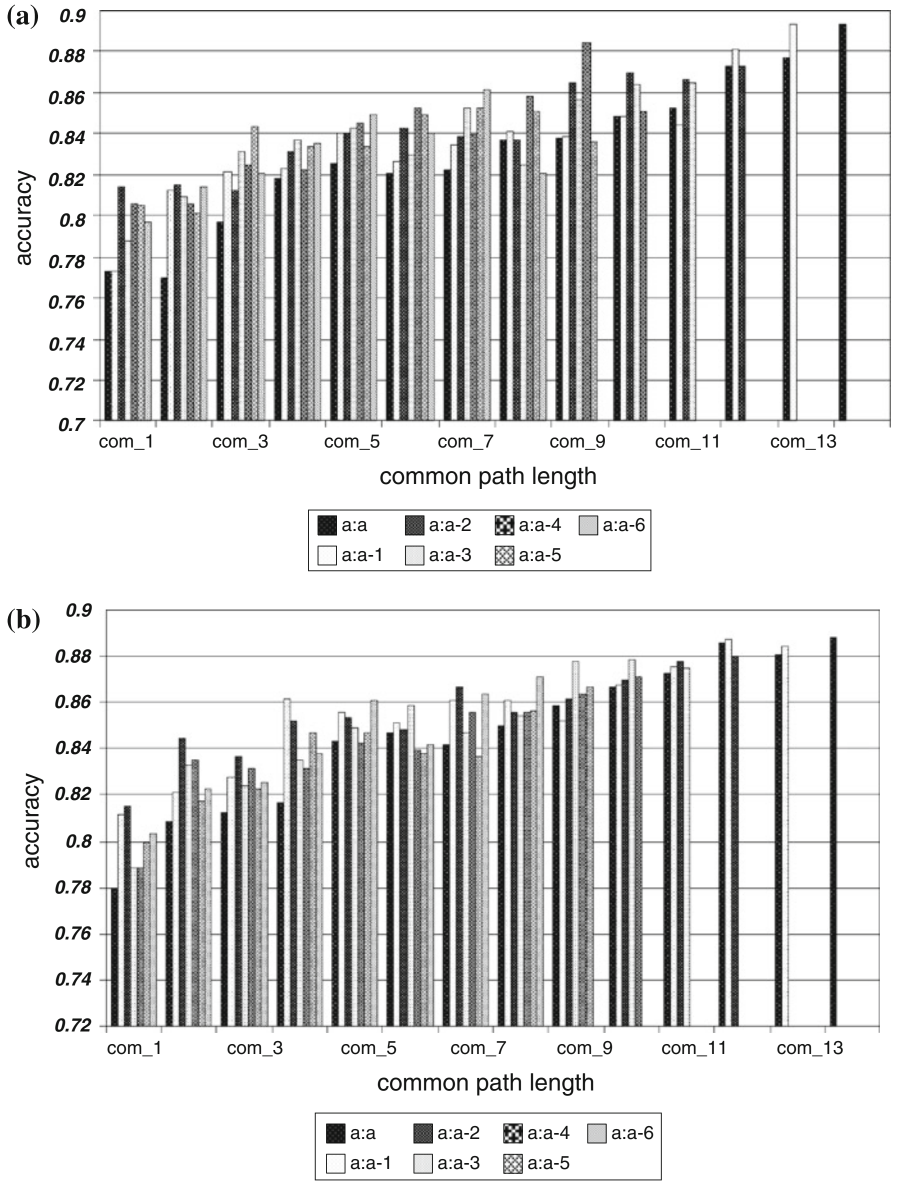
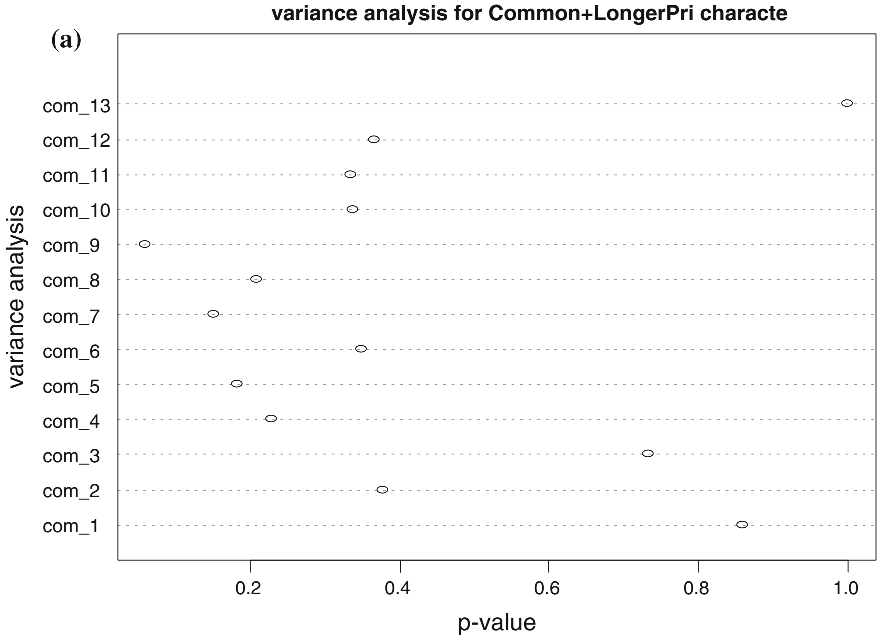


Fig. 4 Relationship among accuracies for the LECs that have the same common and longer private path lengths

Figure 5 shows that almost all the p values are far above the threshold of 0.05. We can therefore conclude that different shorter private path lengths won't greatly affect accuracy. We call this property of accuracy the "Longer Path" characteristic, which states that for any given LOC and for LECs in which the common and the longer private path lengths are the same, accuracies are not significantly different from each other by varying the shorter private path length.



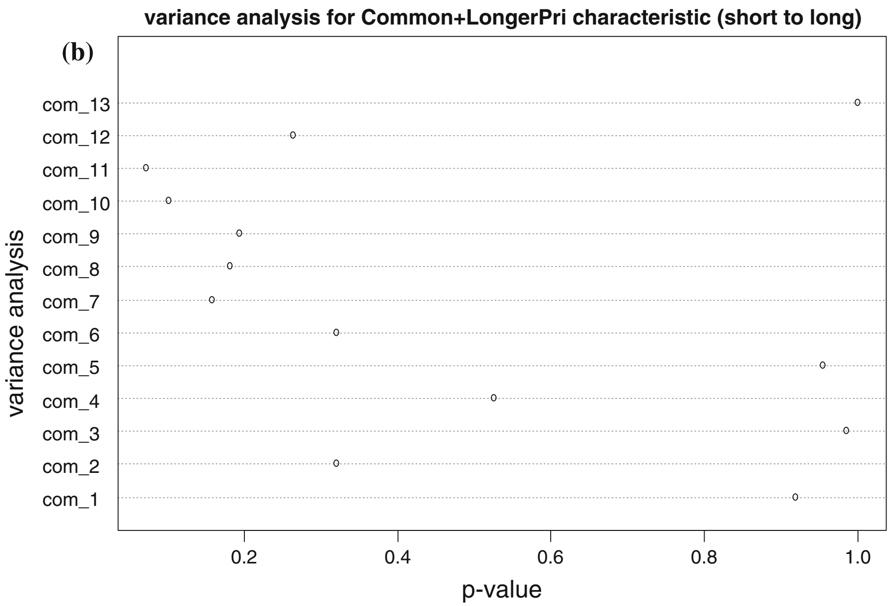


Fig. 5 Variance analysis for the “Longer Path” characteristic

5.2 Minimum Accuracy Characteristic

Although variance analysis shows that there is no significant difference between accuracies of LECs in which only the length of the shorter private path varies, they are still not the same. We now perform further analysis to see how the shorter private path length affects accuracy to identify representative LECs.

We define a “Longer Length” set ($\mathrm{Set_{CLL}}$ for short) as the set of LECs in which the common and the longer private path lengths remain the same, that is,

$$\mathrm{LEC}(x_i, y_i, z_i) \in \mathrm{Set_{CLL}}(c, l) \text{ iff } x_i = c \text{ and } \max(y_i, z_i) = l,\ 1 \le i \le |\mathrm{Set_{LEC}}| \quad (9)$$

Note that $|\mathrm{Set_{LEC}}|$ is the total number of such LECs. With this definition, we apply more analysis on the arrays introduced earlier. Here, we extract 3 two-dimensional arrays, corresponding to the 3 LOCs. The rows of each array are the private path length combinations in which the number of non-zero value items ranges from 1 to 7 depending on the corresponding common path length and the columns are the 13 common path lengths. Again, we are only interested in common path lengths of up to 13, for there is only one accuracy value when the common path length is 14 and, consequently, no comparison can be made.

In each of the 3 two-dimensional arrays, for every common path length, we arrange the accuracy values in the column in an ascending order and assign a value (starting from 1) to an accuracy that corresponds to its place in the ordering. Consequently, there are at most 13 values based on such orderings for any private path length combination. We then form 3 two-dimensional arrays corresponding to the above 3 arrays in which accuracy values are replaced with the ordering values. We further average out all the ordering values from the 3 arrays for the same array entry and the results are listed in Table 1 in which the last column is the mean for each row (the average of the values for a private path length combination over all the 13 common path lengths). We can see from the table that the private path length combination “a:a” has the smallest mean value and, correspondingly, the worst accuracy. We can thus conclude that for any given common path length, among all private path length combinations, accuracy is the worst when the two private path lengths become the same. Although the above analysis has been performed through varying the shorter private path length, the same can be concluded for the reverse case. We call this property of accuracy the “Minimum Accuracy” characteristic.

5.3 Length Ratio Characteristic

Based on the conclusions that we have reached, we now examine accuracy for dynamic path configurations, i.e., for LECs that have the same length ratio that is defined as the ratio of the common path length to the longer private path length. For ease of such an analysis, let’s express LECs as (x, y, z_1) and (n*x, n*y, z_2) in which x, y and z_1 and z_2 represent the lengths of the common path, the longer private path and the shorter private paths, respectively, and define the length ratio r as follows:

$$r = \frac{H_{com}}{\max(H_{pri\_a}, H_{pri\_b})},\ 1 \le H_{com}, H_{pri\_a}, H_{pri\_b} \le 14 \quad (10)$$

Table 1 Values for the "Minimum Accuracy" characteristic

	Com_1	Com_2	Com_3	Com_4	Com_5	Com_6	Com_7
a:a	1.000000	1.666667	1.333333	1.000000	1.666667	2.000000	1.666667
a:a–1	3.000000	1.666667	2.000000	2.333333	2.000000	2.333333	3.000000
a:a–2	2.333333	2.666667	3.666667	2.666667	3.333333	3.000000	3.333333
a:a–3	4.666667	4.000000	4.000000	4.000000	3.333333	4.000000	2.666667
a:a–4	4.333333	5.000000	4.666667	5.666667	4.666667	4.000000	5.333333
a:a–5	6.000000	6.333333	5.333333	5.666667	6.666667	5.666667	6.000000
a:a–6	6.666667	6.666667	7.000000	6.666667	6.333333	7.000000	6.000000

	Com_8	Com_9	Com_10	Com_11	Com_12	Com_13	Mean
a:a	3.000000	2.666667	2.333333	1.333333	2.333333	1.666667	1.666667
a:a–1	2.333333	3.000000	2.666667	1.666667	2.000000	1.333333	2.370370
a:a–2	2.333333	3.333333	3.333333	4.000000	1.666667	X	2.866667
a:a–3	4.333333	3.000000	2.000000	3.000000	X	X	3.484848
a:a–4	6.333333	4.333333	4.666667	X	X	X	4.194444
a:a–5	4.000000	4.666667	X	X	X	X	4.589744
a:a–6	5.666667	X	X	X	X	X	5.179487

That is, r is the ratio of the common path length to the longer private path length. Since the length of any path is at most 15, we deduce that there are a total number of 71 length ratios. Let's further define length ratio set R_r as one that includes all the LECs that have the same length ratio, that is,

$$\mathrm{LEC}(x_i, y_i, z_i) \in R_r \text{ iff } x_i/y_i = n,\ 1 \le i \le |\mathrm{Set}_{\mathrm{LEC}}| \tag{11}$$

From our experiment data, accuracies for LECs that have the same length ratio can be derived, as shown in Fig. 6, that include two estimation cases. For purpose of clarity, only the results that correspond to integer length ratios are presented. Since the length of any complete path is at most 15, there are at most 11 integer length ratios, ranging from 1/6 to 6/1, as shown in Fig. 6 in which "x–y" denotes a length ratio and "nc" denotes an LEC within a length ratio set. We can see from the figure that the accuracies are very close to each other for LECs that have the same length ratio. Figure 6 also shows that the length ratio set R_r corresponding to n = 1 has the worst accuracy and that the higher the length ratio is, the better the accuracy tends to get.

We also perform variance analysis to support the results shown in Fig. 6. First, we construct a three-dimensional matrix of 11*3*7 in which the first dimension denotes the 11 length ratios, the second dimension denotes the 3 LOCs and the third dimension denotes the maximum number of cases for each length ratio which is 7. Second, we extract the following from the three-dimensional matrix: 11 two-dimensional arrays each of which has 3 rows corresponding to the 3 LOCs and 2 to 7 columns corresponding to the number of cases for the respective length ratio. Third, we perform variance analysis for each two-dimensional array using the Binomial test when there are only 2 cases for the length ratio and the Friedman test

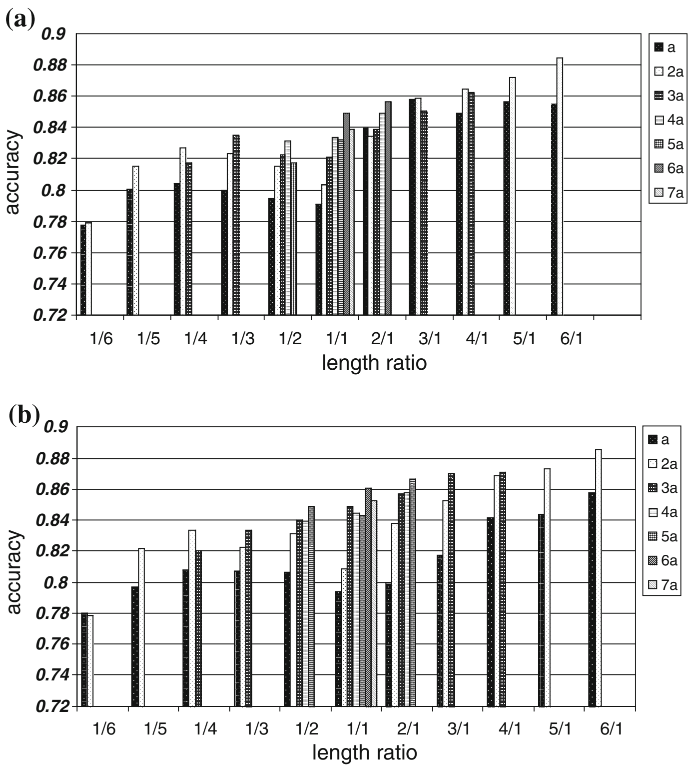
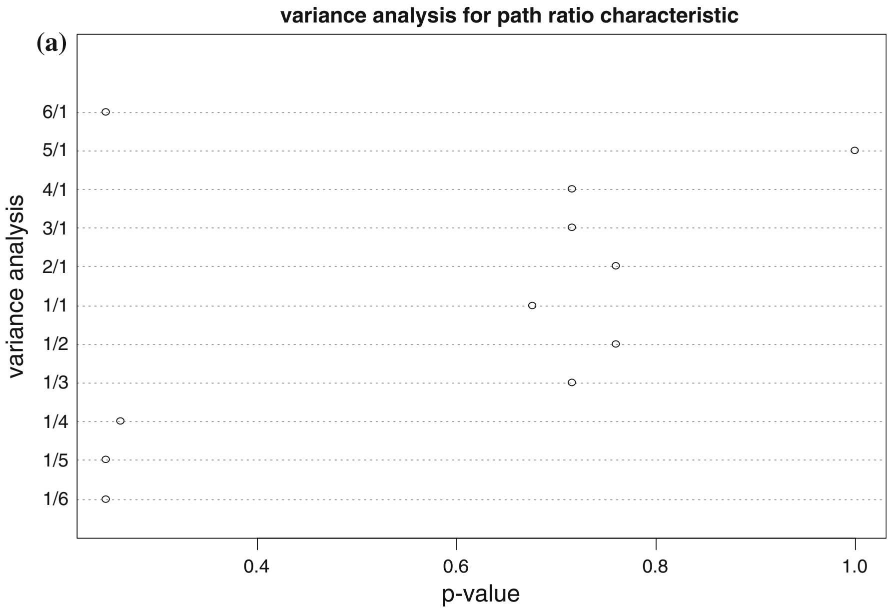


Fig. 6 Accuracies for different length ratios

when there are 3 or more. Figure 7 shows the p values on the analysis from which we can see that all the p values are above the threshold value 0.05 (the p value for length ratio 1/6 is 0.25). Therefore, accuracies for LECs with the same length ratio have an insignificant difference to each other. We call this property of accuracy the “Length Ratio” characteristic. And the higher the length ratio, the better the accuracy tends to get.

5.4 Relationship between Accuracy and Kendall Coefficient

Based on the conclusions we have arrived at so far, we can perform the analysis between accuracy and length ratio. Figure 8a, b, c show the overall 763 accuracies with 763 Kendall coefficients for each of the 3 LOCs from which we derive a linear relationship between the Kendall coefficients and accuracies. We should note that the number of Kendall, as well as accuracy, is not the same in every length ratio, so the distribution of accuracies in every length ratio may affect the overall linear regression for Kendall and accuracy. So, we choose a representative accuracy and its corresponding Kendall for each length ratio in such a manner that every path ratio has the same weight in the linear regression. From the “Longer Path”, the “Minimum Accuracy” and the “Length Ratio” characteristics, we can use the



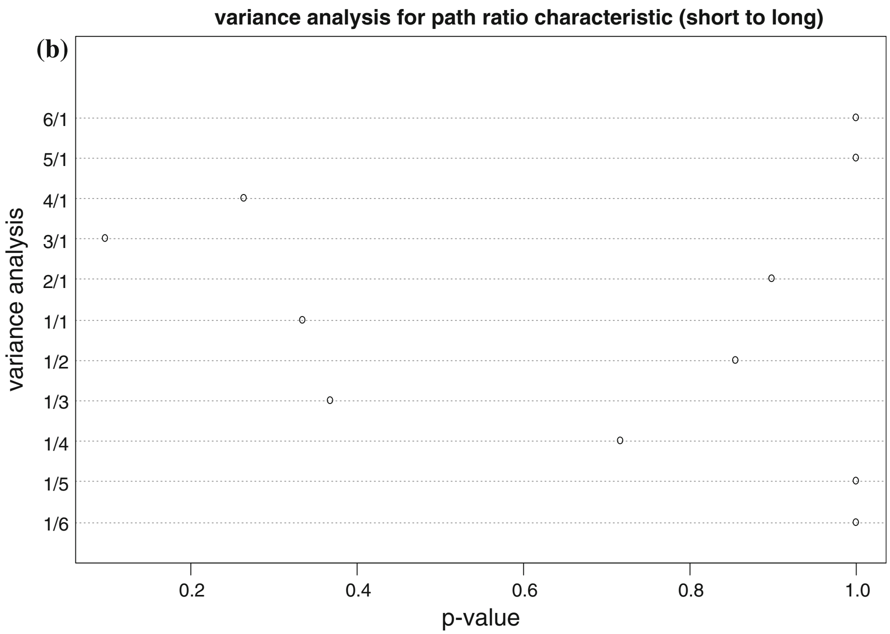


Fig. 7 Variance analysis for the "Length Ratio" characteristic

accuracy of LEC (x, y, y) to represent all the accuracies for LECs of the form (nx, ny, z) where $z < ny$ and $n > 1$. By using representative accuracies, we can perform linear regressions and the results are shown in Fig. 8c, d, e.

5.5 Accuracy Estimation based on Length Ratios

Since there exists a linear relationship between accuracy and the Kendall coefficient, we can approach accuracy by using the Kendall coefficient. Based on our previous study on the relationship between the delay correlation and length ratio using Kendall [13], we can simply use the Kendall vector that contains an ordered list of length ratios. According to the linear regression equation, we can obtain the accuracy for each and every of the 27 length ratios and the results are shown in Fig. 9 which are based on the accuracy that we have for each and every LEC and LOC. For example, when the network load is 80–80–80 %, if we perform the delay estimation for a shorter path based on the delay value for a longer path, we can see that when the length ratio changes from 1/14 to 1/4, accuracy is almost always lower than 0.7. When the length ratio changes from 1/3 to 1/1, accuracy varies between 0.7 and 0.75. When the length ratio is 6/1 or higher, the accuracy becomes higher than 0.8. Furthermore, when the length ratio goes beyond 9/1, accuracy will achieve 0.9.

We can also cluster length ratios together based on their corresponding relative errors. As a result, three clusters emerge, which are (1) when the length ratio changes from 1/14 to 1/1, accuracy varies between 0.62 and 0.85, (2) when the length ratio changes from 2/1 to 4/1, accuracy varies between 0.74 and 0.88, and (3) when the length ratio changes from 5/1 to 14/1, accuracy varies between 0.81 and 0.9.

In the other scenario in which delay estimation is performed for a longer path based on the delay value for a shorter path, (1) when the length ratio changes from

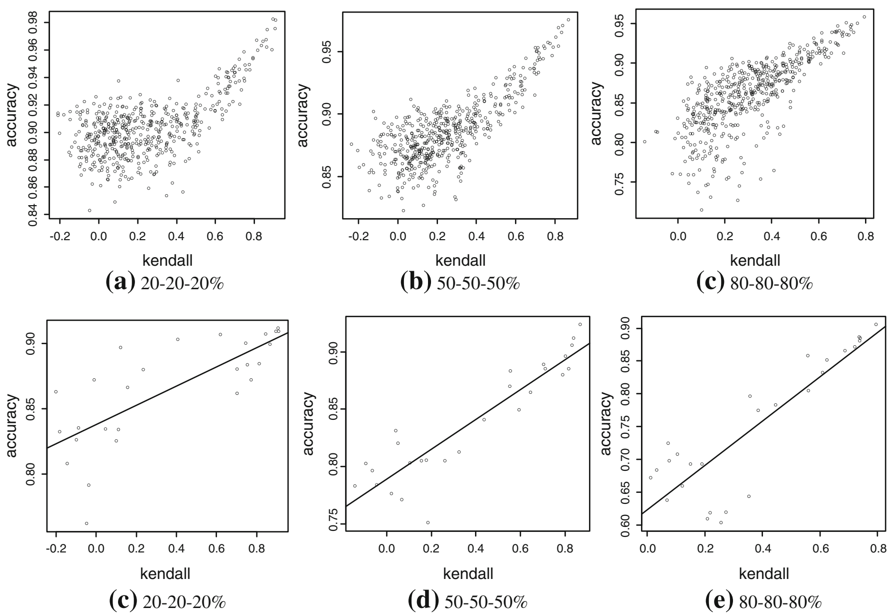


Fig. 8 Relationship between accuracy and Kendall coefficient

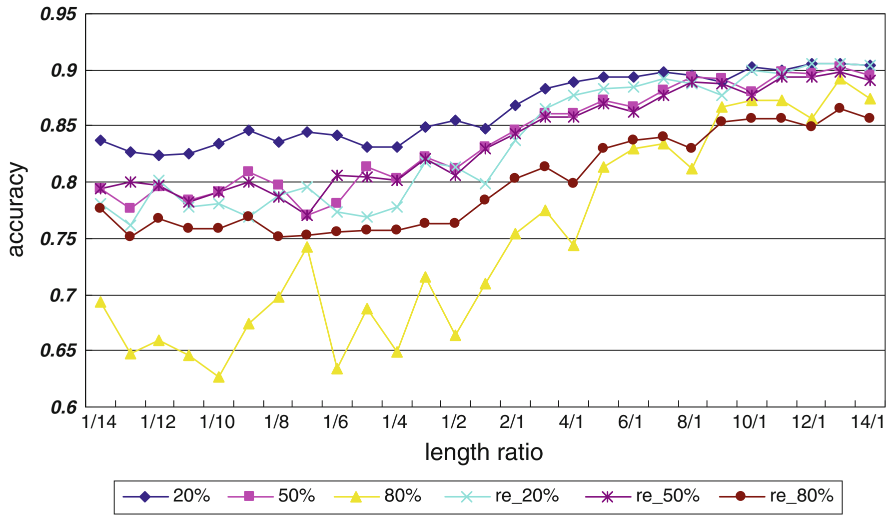


Fig. 9 Accuracy for different length ratios

1/14 to 1/1, accuracy varies between 0.75 and 0.83, (2) when the length ratio changes from 2/1 to 4/1, accuracy varies between 0.8 and 0.87, and (3) when the length ratio changes from 5/1 to 14/1, accuracy varies between 0.83 and 0.9. So, if we require the accuracy to be at least 0.7, we should keep the length ratio higher than 2/1. And the length ratio should be at least 5/1 if the accuracy is required to be at least 0.85.

6 Evaluation with Real Network Data

In this section, we present the results of our work on the evaluation of our proposed delay estimation method based on real RTT data from the Internet in which we use Formula (5) to calculate the accuracy.

6.1 The Data Set

In the evaluation, we used the Rocketfuel data set [22] obtained from a measurement network with 33 monitor nodes on the Planetlab [23] that probed over 120 K destinations, which is one of the most comprehensive and publicly available RTT data sets. We processed the data to suit the needs of the evaluation. First, we deleted 3 monitor probes from the 33 because they could not produce valid traceroute results for a large portion of the destinations, leaving us with 30 monitor nodes. Second, although the original data set that we used covered 3 days, to limit the amount of work, we only used the data from one of the 3 days in the evaluation as we believed that it would be sufficient to evaluate our method. Third, as the destination nodes were randomly selected based on the BGP table entries during that day, some destination IP addresses did not necessarily correspond to the online hosts and, hence, some traceroutes might not have been completed. Therefore, we chose the destination nodes that could be reached by at least 28 monitor nodes and, as a result, got 5,386 destinations nodes. We then put the probe data in 30 groups, each of which corresponded to the measurement results for one of the 30 monitor nodes. The traceroute results to all the 5,386 destinations from this monitor node became the elements of this group. At last, we measured and estimated delays between 435 (i.e., 30*29/2) pairs of source nodes and 5386 (or less) common destination nodes.

6.2 The Results

To evaluate the accuracy of our delay estimation method, we used the real data set to compare the estimation accuracy calculated using Formula (5) with the original relative difference calculated using Formula (3). If the results show that the estimation accuracy is somehow always higher than the original relative difference, then we can say that our method for delay estimation is valid. Note that when using Formula (5), the Kendall coefficient T is related to the path ratio according to [14]. For the 435 pairs of source nodes and approximately 5,368 common destination nodes, we calculated the path ratios as the round up ratio of the common path length to the longer private path length so that we could get more valid data to evaluate our method. For example, 78/67 is not a valid path ratio in our simulation study, which only includes path lengths of 15 hops or less. By rounding up 78/67–2, we could then include this path ratio in our experimental path ratios with the corresponding Kendall coefficients. Using this way of handling the real network data in our evaluation, we were able to obtain the results of the estimation accuracy based on path ratios and the original accuracy differences that are shown in Fig. 10. Our study indicates that the total number of path ratios are 5, based on round-up numbers, and the number of estimation accuracy is therefore 10, for there are two possible directions for the estimation.

We can see from Fig. 10a, c, e that the estimation accuracies calculated using Formula (5) are somewhat lower than the respective original accuracies calculated using Formula (3). Therefore, our method of estimating accuracy based on path length ratios can be validated, although the results from the estimation are less accurate than those from the actual network measurement. This is because our method is based on estimation rather than actual measurement with the benefit of reduction in the overall network overhead that resulted from obtaining delays between network nodes.

To validate our RTT estimation method, we compared the relative difference of our method with other alternative delay estimation methods such as IDMaps, King, ISO-bar, etc. In those methods, there is barely any delay estimation because the estimated delay value for a source node is simply the same as the measured delay value of a selected (and presumably correlated) node to the same common destination node, i.e., $Y = X$. In our method, however, we made some adjustments based on the correlation of the two source nodes according to their topology relationships, i.e., $Y = aX + b$, as well as the coefficients that are related to correlations. The results demonstrated in Fig. 10b, d, f clearly show that the relative difference is somewhat less than the other methods, especially when the relative

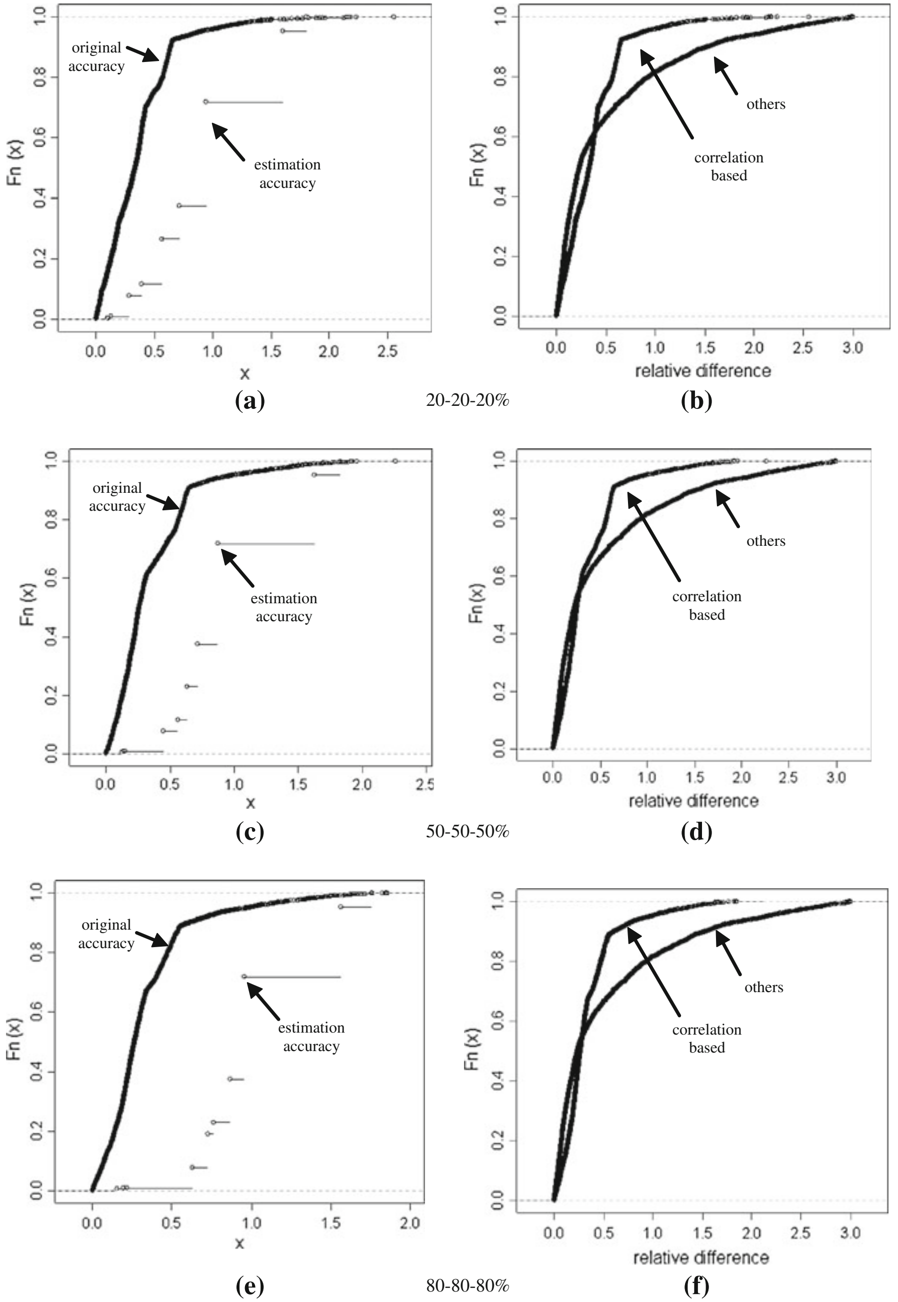


Fig. 10 Evaluation using real network data with loads 20, 50 and 80 %, respectively

difference is higher than 0.4, and our estimation method is noticeably superior to the other alternative methods. Therefore, we can say that our adjustment to the coefficients a and b in delay estimation can be justified and our delay estimation method is effective and can improve delay estimation under various network traffic situations.

7 Framework, Application and Example

In this section, we apply the result of our study on the accuracy of delay estimation to three application scenarios to demonstrate the usefulness and effectiveness of the research results.

7.1 Framework for Delay Measurement and Estimation

In our general framework for delay measurement and estimation, there are at least three types of nodes: the server node, the measurement nodes, and the regular nodes. The server node is responsible for collecting and maintaining measurement results provided by the measurement nodes and for offering delay estimation service to the regular nodes on demand based on measurement results available at the point in time. Measurement nodes could be special monitor nodes in the network or just some regular nodes that are equipped with a delay measurement capability and can thus participate in delay measurement when requested by the server. If delay measurement is performed using a means of software, any network node can be dynamically turned into a measurement node by the server node to meet the needs of delay measurement. Finally, regular nodes are just traditional network nodes and, when need arises, would send a request to the server node for a delay estimation value to a designated destination node. Since delay estimation is performed based on topological properties between network nodes, the server also needs to maintain the topological information on the network, i.e., how nodes in the network are connected to each other.

If measurement nodes are a special type of monitor nodes, then these nodes would serve as service nodes and perform delay measurement to other destination nodes that are regularly in the network. The measurement results are provided to the server node to be used by the server to do delay estimation for other regular network nodes. When a regular network node needs to know the delay between itself and a destination node, it would send a request to the server node for an estimated delay value to the designated destination node. After receiving the request, the server node will find a monitor node that has the best topological correlation to the requesting node. It will then use the measurement results from this monitor node to do delay estimation. The delay estimation result will finally be sent to the requesting node along with an accuracy value for the result. Note that it could be very possible that there would not be any monitor node that exhibits any topological correlation to the requesting node. In this case, no measurement result in the server can be directly used in delay estimation. Thus, in a network in which monitor nodes are sparsely deployed, delay estimation based on a topological correlation of nodes may be feasible only for a subset of the network nodes. This would be the case unless the monitor nodes are strategically located at critical points in the network so that the required topological properties exist between any regular network node and at least one of the monitor nodes.

If delay measurement is implemented using a means of software, any network node can turn into a measurement node, or a monitor node, after the installation of the measurement software in addition to performing its normal functions. Then, measurement functionality can be turned on dynamically by the server node in response to particular needs. If a network node needs a delay estimation result but the server responds with a negative answer, the network node can turn on the delay measurement functionality and perform actual delay measurement whose result will also be sent to the server to be maintained there for other delay estimations. As a result, the server maintains a database of measurement results from network nodes and will use the results in delay estimation. Of course, measurement values may become too old to be useful and those that are obsolete will be purged. In this architecture, delay measurement can be performed very dynamically and delay estimation is used as a mechanism for reducing the overhead brought about by delay measurement in response to the requirement of delay measurement and estimation.

Of course, there could be different scenarios and types of services around which network delay measurement and estimation architecture can be designed. There could also be cases in which only some of the network nodes are equipped with a delay measurement capability, which can be viewed as a general situation that falls between the two more familiar cases illustrated above. Methods of combining the strategies in these cases can be devised, which is out of the scope of this paper and would be part of our future work.

7.2 Application Scenarios

There are different ways in which delay estimation can be useful as the complement to delay measurement in order to reduce network overhead resulting from actual measurement. These scenarios could include delay estimation with required accuracy, full coverage of delay estimation, and selection of the best measurement node.

In the delay estimation with required accuracy scenario, we have a source node M and a destination node D. M performs an actual delay measurement to D. Now, given a required accuracy $A_{require}$, we would like to find out the set of nodes for which when delay estimation is performed for any node in this set based on the measured value from M, the estimated delay values can satisfy $A_{require}$. Consequently, given an accuracy requirement, M can be a representative for this set of nodes and perform an actual delay measurement. All the other nodes in this subset can share the measurement result from M and derive estimated delay values that all meet the accuracy requirement.

In the full coverage of the delay estimation scenario, we have a source node M and a destination node D. Now, we would like to estimate delays from all the "neighboring" source nodes to D based on the measured delay value from M. By applying our analysis result, we will be able to label each estimated delay value for a "neighboring" source node with an accuracy based on the topological relationship of this node to M. Consequently, when the delay measurement result becomes available from M, each "neighboring" node to M can derive an estimated delay value with accuracy. Depending on the requirement of applications on the accuracy of the delay values, some of the estimated values can be used in lieu of performing an actual delay measurement, leading to a reduction in measurement overhead.

In the selection of the best measurement node scenario, a source node can choose from several "neighboring" measurement nodes and use the measured delay value that can give it an estimated delay value with a stated accuracy. Suppose the source node is S, the destination node is D, and the set of the measurement nodes is $\{M_i\}$. If there is an accuracy requirement, we can derive the subset of the measurement nodes that (when being used by S for delay estimation) can satisfy the accuracy requirement due to the topological properties of S and $\{M_i\}$ and the resulting correlation between delays. S can certainly choose from one or more different estimated delay values or choose the best one that enables S to derive an estimated delay value with the highest accuracy.

Note that more application scenarios could be developed as we further refine the analysis on delay estimation. For example, we can propose and solve an optimization problem in which given a network of nodes, the goal would be to determine the minimal set of nodes that (when being used as the delay measurement nodes) could give the best overall accuracy for delay estimation for the whole network. We are currently defining the problem and researching the general approach to deal with this problem.

7.3 An Example

We now use an example to show how to apply delay estimation to delay measurement. The network topology for the example is shown in Fig. 11.

In the example where the topology is illustrated in Fig. 11, node S performs all actual delay (RTT) measurement to node D. Using our analysis result, we estimate the delay (RTT) for the other 7 labeled nodes, i.e., nodes 0–6, to the same node D and compare the estimated values to the actual measurement values to show the accuracy of delay estimation.

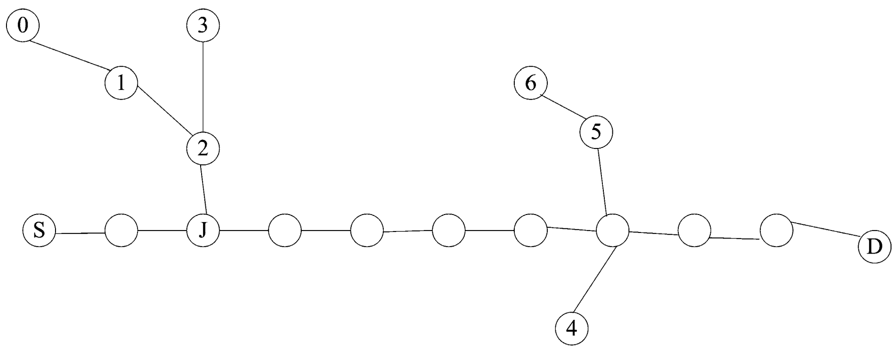


Fig. 11 Network topology for the example

Table 2 Estimation accuracy for all the labeled nodes in the example

	Node 0	Node 1	Node 2	Node 3	Node 4	Node 6	Node 7
Accuracy (load = 20 %, L–S)	0.943168	0.878491	0.865215	0.932289	0.838090	0.891437	0.898399
Overall accuracy (load = 20 %, L–S)	0.883623	0.884434	0.896323	0.889434	0.853823	0.853823	0.836890
Accuracy (load = 50 %, L–S)	0.933101	0.923858	0.902716	0.934803	0.861420	0.837167	0.873893
Overall accuracy (load = 50 %, L–S)	0.868678	0.879383	0.865745	0.879383	0.791246	0.791246	0.829082
Accuracy (load = 80 %, L–S)	0.930317	0.93232	0.911829	0.921017	0.822425	0.803585	0.825958
Overall accuracy (load = 80 %, L–S)	0.810070	0.785288	0.843424	0.785288	0.708030	0.708030	0.677815
Accuracy (load = 20 %, S–L)	0.925915	0.895324	0.899820	0.929940	0.804024	0.824333	0.835947
Overall accuracy (load = 20 %, S–L)	0.883623	0.876899	0.889618	0.876899	0.811159	0.811159	0.779898
Accuracy (load = 50 %, S–L)	0.912649	0.908340	0.907070	0.927064	0.825179	0.821061	0.834268
Overall accuracy (load = 50 %, S–L)	0.868678	0.875660	0.862508	0.875660	0.790666	0.790666	0.827153
Accuracy (load = 80 %, S–L)	0.911868	0.937193	0.960808	0.937721	0.666021	0.666021	0.846384
Overall accuracy (load = 80 %, S–L)	0.810070	0.817708	0.843489	0.817708	0.783446	0.783446	0.770047

We get the 8 RTT sequences, i.e., from node S and the other 7 nodes to node D, respectively, through delay measurement. We also perform delay estimation for the 7 labeled nodes based on the topological relationship between the 7 nodes and S and the measurement result from S to D. We then use Eq. (3) to get the accuracy for the estimated results and the results are listed in Table 2, in which the first six rows show the results for estimating delay over a shorter path using that over a longer path. The second six rows show the opposite case, i.e., estimating delay over a longer path using that over a shorter path for different network loads. Furthermore, for each case, we list the results for the particular node and those for all the nodes that have the same topological relationship to illustrate the difference between the performance for a single instance and that for the overall situation.

8 Conclusions

In this paper, we studied the correlation between packet delays over two paths that have the same destination node. Based on such a correlation, we analyzed the level of accuracy for delay estimation for one path based on the delay measurement result for the other path and showed that the topological properties of the two paths affect accuracy for delay estimation. We performed further analysis and deduced some main characteristics for the accuracy for delay estimation. The significance of our work is that delay estimation can provide results with required accuracies. Then, delay measurement can be performed in such a manner in which measurement nodes can be dynamically selected and other nodes can share measurement results through delay estimation to reduce measurement overhead while supporting the dynamic requirements of delay measurement. We also applied delay estimation to three application scenarios to demonstrate the usefulness and effectiveness of delay estimation based on the application requirements.

Network delay measurement and estimation is very important, although complicated. Hence, there are some limitations in our current study. First, since our study is based on topological properties of the network, it requires that full network topology information be available although there are a number of tools that can be used to collect such information. There are also some other factors that affect delay measurement and estimation. For example, path load, or network load in general, must be considered in delay measurement and estimation. Determining network load at the time of estimation to derive more accurate results appears to be challenging as load measurement (or bandwidth measurement) is another difficult issue that is also under extensive research. Second, if there are only a limited number of measurement nodes, delay estimation may only be applied to a subset of the network nodes that exhibit topological relationships with some of the measurement nodes, i.e., requiring some path commonality between the measurement node and estimation node to a common destination node. Other network properties could be explored to overcome this limitation.

In the future, we will perform further experiments to verify the main characteristics we have derived and apply the results to real network data to further verify and improve the theory and methods for delay estimation. We will also study other problems in delay estimation, such as the optimization problem, to cover all relevant problems for delay estimation.

Acknowledgments The work presented in this paper has been partially supported by National Natural Science Foundation of China (Grant #61272500), Beijing Education Commission Science and Technology Fund (Grant #KM201010005027), and Ministry of Science and Technology of China Public Sector Special Research Fund (Grant #201310040-03). The authors would also like to thank the anonymous reviewers for their valuable comments and suggestions that have been of great help in improving the quality and the presentation of this paper.

References

1. Francis, P., Jamin, S., Jin, C., Jin, Y., Raz, D., Shavitt, Y., Zhang, L.: IDMaps: A global Internet host distance estimation service. IEEE/ACM Trans. Netw. **9**(5), 525–540 (2001)
2. Gummadi, K.P., Saroiu, S., Gribble S.D.: King: Estimating latency between arbitrary Internet end hosts. In: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement, pp. 5–18 (6–8 Nov 2002)
3. Leonard, D., Loguinov, D.: Turbo King: Framework for large-scale Internet delay measurements. In: Proceedings of IEEE INFOCOM 2008, pp. 31–35 (13–18 April 2008)
4. Hariri, N., Hariri, B., Shirmohammadi, S.: A distributed measurement scheme for Internet latency prediction. IEEE Trans. Instrum. Meas. **60**(5), 1594–1603 (2011)
5. Sharma, P., Xu, Z.C., Banerjee, S., Lee, S.J.: Estimating network proximity and latency. ACM SIGCOMM Comput. Commun. Rev. **36**(3), 39–50 (2006)
6. Agarwal, S., Lorch, J.R.: Matchmaking for online games and other latency-sensitive P2P systems. In: Proceedings of ACM SIGCOMM 2009, pp. 315–326 (17–21 Aug 2009)
7. Xing, C.Y., Chen, M.: A virtual node based network distance prediction mechanism. In: Proceedings of IEEE GLOBLECOM 2008, pp. 1–5 (Nov 30–Dec 4 2008)
8. Cong, L., Chen, Y., Shi, X., Wang, Y., Wu, W., Deng, B., Li, X.: Telemeter: network distance prediction based on network topology. In: Proceedings of the 6th International Conference on Grid and Cooperative Computing, pp. 34–41 (16–18 Aug 2007)
9. Elmokashfi, A., Kleis, M., Popescu, A.: NetForecast: a delay prediction scheme for provider controlled networks. In: Proceedings of IEEE GLOBECOM, pp. 502–507 (2007)
10. Lim, H., Hou, J.C., Choi, C.H.: Constructing Internet coordinate system based on delay measurement. In: Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement, pp. 129–142 (27–29 Oct 2003)
11. Tang, L.Y., Crovella, M.: Virtual landmarks for the Internet. In: Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement, pp. 143–152 (27–29, Oct 2003)
12. Xing, C.Y., Chen, M.: Characteristics of the Internet latency and their impact on distance prediction accuracy. In: Proceedings of the 2009 7th Annual Communication Networks and Service Research Conference, pp. 171–177 (11–13 May 2009)
13. Chen, Y., Lim, K.H., Katz, R.H., Overton, C.: On the stability of network distance estimation. ACM SIGMETRICS Perform. Eval. Rev. **30**(2), 21–30 (2002)
14. Zhu, N., He, J.: Experimental study of the similarity of network nodes based on RTT. In: Proceedings of the 2010 International Conference on Internet Technology and Applications. (20–22 Aug 2010)
15. Chen, M.: Opnet Network Simulation. Tsinghua University Press, Beijing, China (2004)
16. Fei, A., Pei, G., Liu, R., Zhang L.: Measurements on delay and hop-count of the Internet. In: Proceedings of IEEE GLOBECOM 1998—Internet Mini-Conference. (8–12 Nov 1998)
17. Legend, W.E., Taqqu, M.S., Willinger, W., Wilson, D.V.: On the self-similar nature of ethernet traffic (extended version). IEEE/ACM Trans. Network. **2**(1), 1–15 (1994)
18. Leys, P., Potemans, J., Van den Broeck, B., Theunis, J., Van Lil, E. Van de Capelle A.: Use of the raw packet generator in OPNET. In: Proceedings of OPNETWORK 2002. (26–30 Aug 2002)
19. Lapata, M.: Automatic evaluation of information ordering: Kendall's Tau. Comput. Linguist. **32**(4), 471–484 (2006)
20. Lan, K.C., Heidemann, J.: A measurement study of correlations of Internet flow characteristics. Comput. Netw. **50**(1), 46–62 (2006)
21. Xue, Y.: Statistical Modeling and Software. Tsinghua University Press, Beijing (2008)

22. Rocketfuel. http://www.cs.washington.edu/research/networking/rocketfuel/
23. PlanetLab. http://www.planet-lab.org/

Author Biographies

Nafei Zhu received her B.S. degree in Communication Engineering and her M.S. degree in Traffic Information Engineering & Control from Central South University in China and her Ph.D. degree in Computer Science and Technology from Beijing University of Technology in China. She is currently an engineer in the National Application Software Testing Labs in Beijing, China. Ms. Zhu has published several research papers in scholarly journals and international conferences. Her main research interests include measurement and analysis of network performance and security.

Jingsha He received his B.S. degree in Computer Science from Xi'an Jiaotong University in China and his M.S. and Ph.D. degrees in Computer Engineering from University of Maryland at College Park, USA. He is currently a professor in the School of Software Engineering at Beijing University of Technology in China. Professor He has published over 170 research papers in scholarly journals and international conferences and has received nearly 30 patents in the United States and China. His main research interests include information security, measurement and analysis of network performance, and security of wireless ad hoc, mesh and sensor networks.

Yue Zhou is currently the Director of the National Application Software Testing Labs in Beijing, China and has over 10 years of experience in software development and testing. Ms. Zhou is an expert in software quality assurance. She participated in many national projects and software quality assurance programs such as the Beijing Olympic Games in 2008 and the Shanghai World Expo in 2010. She has also co-authored several national standards in areas related to software quality assurance. Her main research interests include the application of information technologies and software quality assurance.

Wei Wang received her B.S. degree in Computer Science and Technology from Tsinghua University in China. She is currently a department manager in the National Application Software Testing Labs in Beijing, China. Ms. Wang has participated in over 100 projects in finance, gaming, petroleum, taxation, irrigation, accounting and public health. She has also taken part in drafting several national standards in the above areas.