社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

オーバレイネットワークにおける経路重複の推定に基づく分散型利用可能帯域計測手法

ディンティエン ホアン† 　長谷川　剛†† 　村田　正幸†

† 大阪大学大学院情報科学研究科　〒565-0871 大阪府吹田市山田丘 1-5
†† 大阪大学サイバーメディアセンター　〒560-0043 大阪府豊中市待兼山町 1-32
E-mail: †{d-hoang,murata}@ist.osaka-u.ac.jp, ††hasegawa@cmc.osaka-u.ac.jp

あらまし　利用可能帯域は，遅延時間やパケットロス率とともにオーバレイネットワークサービスの効率向上において重要な指標である．しかし，他の指標と比べて，利用可能帯域の計測は多量の計測オーバヘッドを必要とする．また，計測頻度を高めることにより，計測精度を向上することができるが，計測する経路が重複している場合には，計測の衝突が発生し，計測負荷の増大や計測精度の低下が問題となる．本稿では，計測衝突を軽減し，計測精度を向上する分散型利用可能帯域計測手法を提案する．提案手法においては，オーバレイノードがオーバレイパスの経路情報に関して必要最低限の情報交換を行い，オーバレイパスの経路重複を検出する．経路重複の状況に基づいて，確率的に計測タイミングを決定することにより，計測衝突を軽減する．さらに，経路が重複するオーバレイパスの計測結果を共有し，エンドツーエンド利用可能帯域計測におけるパラメータ設定に用いることにより，計測オーバヘッドを削減する．性能評価の結果，提案手法を用いることにより計測結果の相対誤差を既存手法に比べておよそ 65% まで削減できることを示す．

キーワード　オーバレイネットワーク，ネットワーク計測，利用可能帯域，計測衝突，情報交換

A distributed method for measuring available bandwidth in overlay networks exploiting path overlap

Dinh TIEN HOANG†, Go HASEGAWA††, and Masayuki MURATA†

† Graduate School of Information Science and Technology, Osaka University　1-5, Yamadaoka, Suita,Osaka, 565-0871 Japan
†† Cybermedia Center, Osaka University　1-32, Machikaneyama-cho, Toyonaka, Osaka, 560-0043 Japan
E-mail: †{d-hoang,murata}@ist.osaka-u.ac.jp, ††hasegawa@cmc.osaka-u.ac.jp

Abstract　The estimation of available bandwidth is crucial for many overlay network applications. However, measuring available bandwidth requires a large amount of probe traffic. Furthermore, the measurement conflict of route-overlapping paths can cause serious degradation of measurement accuracy and non-negligible increase in the network load. In this report, we propose a distributed method for measuring available bandwidth in overlay networks that can reduce measurement conflict while maintaining high measurement accuracy with low cost. The main idea is that neighboring overlay nodes exchange the route information to detect overlapping paths, and share measurement results of overlapping paths to configure parameter settings for available bandwidth measurement. Simulation results show that the relative errors in the measurement results of our method are approximately only 65% of those of the existing method.

Key words　Overlay networks, network measurement, available bandwidth, measurement conflict, information exchange

1 Introduction

Available bandwidth, along with latency and packet loss ratio, is an essential metric for efficient operation of overlay network applications. For example, available bandwidth information allows the construction of an efficient overlay topology for video on demand [1] and peer-assisted streaming [2]. However, measuring available bandwidth in overlay networks is generally expensive, not only because of the huge number of pairwise measurements but also because of the large traffic load of each measurement. In particular, for an overlay network that contains n overlay nodes, the number of pairwise measurements is $O(n^2)$, which becomes unacceptably large in large-scale overlay networks. Furthermore, the traffic load of each measurement of available bandwidth is much larger than that of other metrics such as latency or packet loss rate. This is because latency and packet loss rate can be measured using lightweight tools such as `ping`, whereas measuring available bandwidth requires more complicated and costly mechanisms. For example, in the case of Pathload [3], which is one of the most accurate tools for measuring end-to-end available bandwidth, groups of packet streams are sent at various rates within a large range that contains the real value of available bandwidth. The traffic load of each Pathload measurement is therefore very large, reaching up to 10 MB according to one study [4]. However, most existing solutions [5]〜[7] focus on decreasing the number of pairwise measurements rather than reducing the traffic load of each measurement.

Another measurement issue in overlay networks is measurement conflict, which degrades measurement accuracy. This problem occurs when measurements of overlapping paths are performed simultaneously. Previous studies have addressed this problem, and algorithms for avoiding concurrent measurements of overlapping paths have been proposed [8], [9]. Although measurement conflicts can be completely avoided by using these methods, the measurement frequency is small, thus leading to inaccurate measurement results [10]. Furthermore, concurrent measurements of overlapping paths do not always conflict depending on the mechanisms employed by the measurement tools. For example, in the case of Pathload, because the interval between two consecutive packet streams is set to a value not smaller than one RTT, if the duration of sending a single packet stream is smaller than one RTT, the probability of a conflict occurring is smaller than that of non-conflict.

In this report, we propose a distributed method for measuring available bandwidth that addresses both of the above problems: reducing the measurement traffic load and minimizing the effect of measurement conflicts. Unlike existing solutions [5]〜[7], the approach we take focuses on decreasing the traffic load of each measurement. This approach not only reduces the total measurement traffic load but also helps mitigate measurement conflicts. The proposed method does not completely avoid concurrent measurements of overlapping paths like the solutions in [8], [9], but instead reduces the number of concurrent measurements while maintaining a high measurement frequency to improve measurement accuracy.

In our method, overlay nodes exchange route information in order to detect overlapping paths, as proposed in our previous study [10]. The measurement frequency and timing of each path are determined based on the overlapping state in order to reduce measurement conflicts. To obtain accurate measurement results, we adopt some mechanisms similar to induced-congestion-based end-to-end available bandwidth tools such as Pathload or pathChirp [11] for measuring end-to-end available bandwidth. Measurement traffic load is reduced by having the overlay nodes exchange the measurement results of overlapping paths and then use this information for calculating the parameters for each measurement. We evaluate our method and compare it with a previous method [8] by simulations of both generated and real Internet topologies. The simulation results show that the relative errors in the measurement results of our method are only approximately 65% those of the method from [8].

The rest of this report is organized as follows. Definitions related to overlay networks are presented in Section 2. In Section 3, we explain our method for reducing measurement conflicts and decreasing the traffic load of each measurement. We evaluate our method in Section 4 and give the conclusions of this report in Section 5.

2 Network model and definitions

Consider an overlay network in which the overlay nodes are installed on routers or end hosts. This installation can be done in the networks that support configurations at the application level in the routers. If the network supports such techniques as network virtualization [12] and software defined networks [13], which enable the settings of all network components at some devices called controllers, this installation can be further simplified. Suppose that the network contains m end hosts or routers denoted R_i $(i = 1, ..., m)$. For simplicity, we refer to each end host or router as an *underlay node*. Suppose that n $(n \leqq m)$ overlay nodes, denoted O_i $(i = 1, ..., n)$, are deployed on n different underlay nodes. The density σ of overlay nodes is defined as the ratio of overlay nodes to underlay nodes, that is, $\sigma = n/m$. Figure 1 shows an example of an overlay network. Gray arrows indicate overlay paths and black arrows indicate the underlay paths that correspond to the overlay paths. We assume the shortest path algorithm for routing in the underlay network and define R_iR_j as the underlay path between underlay nodes R_i and R_j, where R_i is the *source node* and R_j is the *destination node* of the path. If different paths R_iR_j and R_sR_t share at least one link, then R_iR_j and R_sR_t overlap and we say that R_iR_j (R_sR_t) is an *overlapping path* of R_sR_t (R_iR_j). We define a *route* from R_i to R_j as a sequence of underlay nodes that construct an underlay path from R_i to R_j.

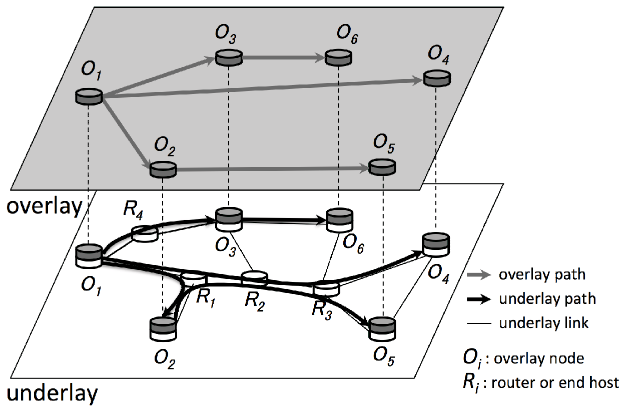


Fig. 1 Example of an overlay network and overlapping paths

As in our previous work [10], we classify the overlapping paths into the following three types:

- Complete overlapping: One path completely includes another path. The path that includes the other path is called the *longer path*, and the included path is called the *shorter path*.
- Half overlapping: Two paths share a route from the source node to a router that is not an overlay node.
- Partial overlapping: Two paths share a route that does not include the source node.

For example, in Fig. 1, path O_1O_3 is a complete overlapping path of O_1O_6. Paths O_1O_2 and O_1O_4 have a half overlapping relation, and path O_1O_4 is a partial overlapping path of O_2O_5.

3 Proposed method

3.1 Overview

Our solution is built in a completely distributed fashion in which each overlay node measures the paths starting from itself based on information obtained by exchanges with neighboring overlay nodes. The measurement procedure employed by each overlay node consists of the following three phases:

- Detection phase of overlapping paths

The overlay nodes detect overlapping paths by using a previously described method [10].

- Calculation phase of measurement timings

The frequencies and timings for measuring each of the paths are calculated based on the type of overlap.

- Measurement phase

At each measurement timing, the overlay node calculates the parameters for the end-to-end measurement based on previous mea-

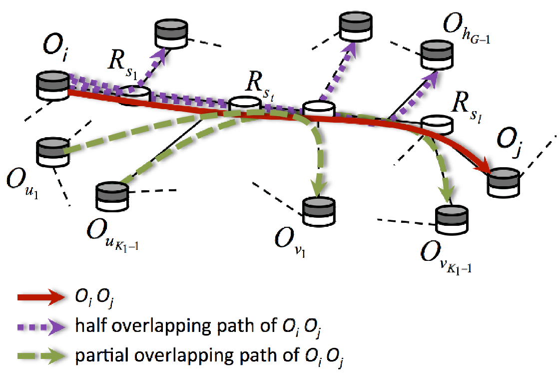


Fig. 2 Example for explaining the proposed measurement method

surement results received from other nodes. The overlay node performs measurement using these parameters, and then sends the results and related information to the neighboring overlay nodes.

3.2 Detection phase of overlapping paths

We use an existing method [10] to detect complete, half, and partial overlapping paths in the overlay network. In particular, an arbitrary overlay node O_i can detect complete and half overlapping paths of path O_iO_j by issuing `traceroute` commands to all other nodes. To detect the partial overlapping paths of O_iO_j, O_i first utilizes the overlapping status of the half overlapping paths to find the candidates of partial overlapping paths and then exchanges the routing information with the source nodes of the candidates to determine their overlapping states. For example, in Fig. 1, we infer that path O_2O_5 is a candidate of partial overlapping path of O_1O_4, because the length of the overlapping part of O_1O_4 and O_1O_2 is smaller than the length of the overlapping part of O_1O_4 and O_1O_5. O_1 then exchanges routing information with O_2 to confirm whether O_2O_5 is actually a partial overlapping path of O_1O_4. Our simulation results indicate that our method can detect approximately 90% of partial overlapping paths with relatively small overhead [10].

3.3 Calculation phase of measurement timings

We propose a method for calculating the measurement timings of the paths that can reduce measurement conflicts while maintaining high frequencies to improve measurement accuracy. Our method utilizes the overlapping states of the paths.

For complete overlapping paths, we only measure the shorter path in order to avoid conflicts. The longer path is not directly measured, and instead the measurement result is estimated based on the measurement results of contained shorter paths [10].

We explain the method for half and partial overlapping paths as follows. Consider path O_iO_j that has half and partial overlapping paths (Fig. 2). We denote by $(G_{i,j} - 1)$ the number of half overlapping paths of O_iO_j $(G_{i,j} \geqq 1)$. For simplicity, we refer to $G_{i,j}$ as G. We refer to path O_iO_j as path 1, and to each of the half overlapping paths as path p $(2 \leqq p \leqq G)$. We then denote by $(K_p - 1)$ the number of partial overlapping paths of path p, with $1 \leqq p \leqq G$ and $K_p \geqq 1$.

Overlay node O_i can avoid measurement conflicts among the half overlapping paths 1, 2, ... G simply by measuring them sequentially. Conflicts between the partial overlapping paths, however, cannot be avoided completely since the source nodes of the partial overlapping paths are different. We therefore propose a technique that combines sequential measurement for half overlapping paths and random measurement for partial overlapping paths. We set the time required to measure a single path to a predetermined parameter τ. We assume that O_i aggregates all of the measurement results of paths 1 to G after a predetermined duration, which we call an *aggregation period*. The aggregation period is divided into T $(T \geqq 1)$ *measurement time slots* of length τ. We denote by h_p the number of times the path p is measured within an aggregation period $(h_p \leqq T)$ and calculate h_p as follows.

Let us introduce β_p as a value that reflects the variability of the measurement results of path p during an aggregation period. Note that the method for determining β_p is beyond the scope of this report. For example, β_p can be calculated based on the statistics of the measurement results or using an existing method [14]. We set the number of measurements h_p to be proportional to β_p among all paths, that is, $h_1/\beta_1 = h_2/\beta_2 = ... = h_G/\beta_G$. To avoid measurement conflicts between half overlapping paths, the sum of the number of measurements should be less than or equal to T: $\sum_{p=1}^{G} h_p \leqq T$. This gives $h_p \leqq T\beta_p/(\sum_{s=1}^{G} \beta_s)$. To reduce measurement conflicts between path p and the $(K_p - 1)$ partial overlapping paths, we set the number of measurements of path p to a value less than or equal to T/K_p, that is, $h_p \leqq T/K_p$. In addition, we want to make the number of measurements as large as possible to obtain as many measurement results as possible. Accordingly, we set $h_p = min\{T\beta_p/(\sum_{s=1}^{G} \beta_s), T/K_p\}$.

Next, we propose Algorithm 1 for allocating the measurement timings of path p in an aggregation period such that the number of measurements of path p becomes h_p. The main idea of the algorithm is to divide the T measurement time slots of an aggregation period into h_p groups, and then randomly choose one slot from each group to allocate to path p.

Algorithm 1 Method for allocating measurement timings

function AllocMeasTime()
for $p = 1$ to G **do**
  Let c $(c \leqq T)$ be the number of slots that have not been allocated to any path
  Divide these c slots into h_p groups, so that each group contains c/h_p continuous slots
  Randomly choose one slot from each group and allocate it to path p
end for
end function

3.4 Measurement phase

In this section, we explain our method that sets the parameters for each end-to-end measurement to reduce the measurement traffic load. Our method can be applied for induced-congestion-based measurement tools such as Pathload or pathChirp [11]. We only present the method for Pathload in the interest of saving space.

3.4.1 Calculating parameters for available bandwidth measurement

To obtain accurate measurement results, we adopt a mechanism similar to Pathload for measuring the end-to-end available bandwidth. However, since the default settings for the parameters in each Pathload measurement result in very large traffic load, we propose a statistical method for calculating these parameters in order to reduce the measurement traffic load.

We first need to understand why Pathload produces large measurement traffic load. Pathload relies on the fact that the one-way delays of a periodic packet stream show an increasing trend when the stream rate exceeds the available bandwidth. It begins with a large range (R_{min}, R_{max}) and uses a binary search algorithm to find the value of available bandwidth within this range. More specifically, at each iteration of a measurement, the source node sends a string of packet streams called a *packet fleet* at the rate $R^* = (R_{min} + R_{max})/2$ and checks whether there is an increasing trend in the one-way delays to judge if the real value of available bandwidth is larger or smaller than this rate. If the real value of available bandwidth is found to be larger than this rate then R_{min} is set to R^* otherwise R_{max} is set to R^* and the search procedure is repeated. Once the width of the search range (R_{min}, R_{max}) becomes smaller than some predefined threshold ω, the procedure stops and (R_{min}, R_{max}) is reported as the measurement result. It is obvious that the traffic load of each measurement depends on the width of the initial search range. Since the initial value of R_{min} is set to 0 and the initial value of R_{max} is set to some large value, for example the capacity of the path, the measurement traffic load is very large [4].

In our method, overlay nodes exchange measurement results of overlapping paths and related information in order to calculate a narrower search range (R_{min}, R_{max}) that is closer to the actual value of available bandwidth, with the aim to reduce the traffic load of each measurement. We rely on the observation that when the tight links of two overlapping paths are in the overlapping part, the measurement result of one path can be used as the measurement result of the other.

More specifically, let us consider a path O_iO_j. We first assume that path O_iO_j has K partial overlapping paths ($K \geqq 1$) denoted $O_{u_s}O_{v_s}$ ($1 \leqq s \leqq K$). O_i receives the following information from each O_{u_s} ($1 \leqq s \leqq K$).

(1) The measurement result of $O_{u_s}O_{v_s}$

(2) The probability that the tight link of $O_{u_s}O_{v_s}$ belongs to the overlapping part of O_iO_j and $O_{u_s}O_{v_s}$, denoted as $\Phi_{O_{u_s}O_{v_s},O_iO_j}$.

We calculate $\Phi_{O_{u_s}O_{v_s},O_iO_j}$ as follows by using an existing method [14]:

$$\Phi_{O_{u_s}O_{v_s},O_iO_j} = \frac{Latency(Overlap(O_iO_j, O_{u_s}O_{v_s}))}{Latency(O_{u_s}O_{v_s})},$$

where $Overlap(O_iO_j, O_{u_s}O_{v_s})$ is the overlapping part of paths O_iO_j and $O_{u_s}O_{v_s}$.

After receiving the above data, O_i also estimates $\Phi_{O_iO_j,O_{u_s}O_{v_s}}$, which is the probability that the tight link of O_iO_j belongs to the overlapping part of O_iO_j and $O_{u_s}O_{v_s}$. It then calculates $\alpha_s = \Phi_{O_iO_j,O_{u_s}O_{v_s}}\Phi_{O_{u_s}O_{v_s},O_iO_j}$, which is the probability that the tight links of O_iO_j and $O_{u_s}O_{v_s}$ belong to the overlapping part of O_iO_j and $O_{u_s}O_{v_s}$. This means that α_s is the probability that the measurement results of O_iO_j and $O_{u_s}O_{v_s}$ are equal.

O_i stores the results of its own measurements as well as the information received from other nodes. This stored data are used to calculate R_{min} and R_{max}, and are discarded when it is determined that the data are no longer useful for the calculation.

We assume that at some measurement timing t^*, O_i has stored G measurement results of O_iO_j and its half and partial overlapping paths, which we denote $(A_L^1, A_U^1), (A_L^2, A_U^2), ..., (A_L^G, A_U^G)$. We then denote by α_1, α_2, ..., α_G the probabilities that each of the corresponding results equals the measurement results of O_iO_j. Note that α_s ($1 \leqq s \leqq G$) corresponding to the measurement result of O_iO_j is set to 1.

We calculate the lower bound of the 95% confidence interval of A_L^s ($1 \leqq s \leqq G$), denoted S_L^*, and the upper bound of the 95% confidence interval of A_U^s ($1 \leqq s \leqq G$), denoted S_U^*, as follows:

$$S_L^* = \bar{A}_L - 1.96\sqrt{\frac{V_L}{G}} \quad , \quad S_U^* = \bar{A}_U + 1.96\sqrt{\frac{V_U}{G}}. \tag{1}$$

Here, $\bar{A}_L$, V_L, $\bar{A}_U$, and V_U are the weighted means and variances, calculated as follows:

$$\begin{aligned} \bar{A}_L &= \sum_{s=1}^{G} \beta_s A_L^s \quad , \quad V_L = \sum_{s=1}^{G} \beta_s {A_L^s}^2 - \bar{A}_L^2, \\ \bar{A}_U &= \sum_{s=1}^{G} \beta_s A_U^s \quad , \quad V_U = \sum_{s=1}^{G} \beta_s {A_U^s}^2 - \bar{A}_U^2, \end{aligned} \tag{2}$$

where $\beta_s = \alpha_s / \sum_{w=1}^{G} \alpha_w$ ($1 \leqq s \leqq G$) is the weight of result (A_L^s, A_U^s).

We infer that the real value of the available bandwidth is either near or within the range (S_L^*, S_U^*) and set $R_{min} = S_L^*$ and $R_{max} = S_U^*$.

34.2 Performing measurement

Since we are not sure whether the real value of available bandwidth is actually within the range (S_L^*, S_U^*), O_i first sends probing packet streams at rates of S_L^* and S_U^* to determine if the real value is between S_L^* and S_U^* based on the presence of an increasing trend in one-way delays. If the real value of available bandwidth is not between S_L^* and S_U^*, we infer that it has changed greatly and discard the stored measurement results because that data has become unreliable. We also infer that the real value exists outside but near the range (S_L^*, S_U^*). We then choose a new search range that neighbors the range (S_L^*, S_U^*) and check whether the real value of available bandwidth is in this new range. This procedure is repeated until we find a search range that includes the real value of available bandwidth. We then apply an algorithm that is similar to Pathload to search for the real value of available bandwidth.

Algorithm 2 Measurement algorithm for path O_iO_j

```
function MeasureOnePath()
// Initialize
R_min ← S*_L
R_max ← S*_U
upper_found ← 0
lower_found ← 0
meas_time ← τ

// Find the range (R_min, R_max) that contains available bandwidth
while (upper_found = 0 || lower_found = 0) && meas_time > 0 do
    if upper_found = 0 then
        Send a packet fleet at rate R_max
        Subtract the time taken to send the packet fleet from meas_time
        if increasing trend then
            upper_found ← 1
        else
            R_min ← R_max
            lower_found ← 1
            R_max ← min(R_max + (S*_U − S*_L)/2, C^0_{OiOj})
        end if
    end if
    if lower_found = 0 && meas_time > 0 then
        Send a packet fleet at rate R_min
        Subtract the time taken to send the packet fleet from meas_time
        if non increasing trend then
            lower_found ← 1
        else
            R_max ← R_min
            upper_found ← 1
            R_min ← max(R_min − (S*_U − S*_L)/2, 0)
        end if
    end if
end while

// Measure available bandwidth in the range (R_min, R_max)
if R_max − R_min > ω && meas_time > 0 then
    RuntimeLimitedPathload(R_min, R_max, meas_time)
end if
return R_min, R_max
end function
```

In Pathload, the search procedure stops when the width of the search range is smaller than the threshold ω. In the proposed method, we add another termination condition to the search procedure, which is to stop if the time taken by the measurement exceeds τ.

The details of our method are shown in Algorithm 2. $C^0_{O_iO_j}$ is the capacity of the first IP link of path O_iO_j. The procedure *RuntimeLimitedPathload* is the a search procedure based on Pathload with limited search time.

After O_i has measured O_iO_j, it sends the result and probabilities $\Phi_{O_iO_j,O_{u_s}O_{v_s}}$ to nodes O_{u_s} ($1 \leqq s \leqq K$).

Assume that during an aggregation period, O_i obtained F measurement results of O_iO_j, denoted as $(A_L^1, A_U^1), (A_L^2, A_U^2), ..., (A_L^F, A_U^F)$. The measurement result of O_iO_j at that aggregation period is calculated by Eq. (3):

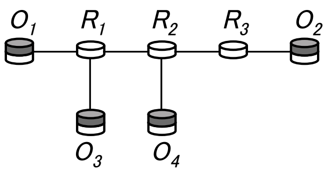


Fig. 3 Small network topology

$$A_{meas} = \frac{1}{F}\sum_{s=1}^{F}\frac{A_L^s + A_U^s}{2}. \qquad (3)$$

4 Performance evaluation

4.1 Evaluation method

We performed simulations to examine whether the proposed method works correctly as designed, and to compare the performance with that of an existing method [8]. Throughout the simulation, we assume that Pathload is used for the end-to-end measurement of the available bandwidth. However, we expect the same trend in the evaluation results in the cases of other measurement tools. In the method from [8], since overlay nodes do not exchange information with each other in this method, the search range for each end-to-end measurement cannot be estimated, unlike our proposed method. We therefore set the search range for path O_iO_j to $(0, C^0_{O_iO_j})$.

We compare our proposed method and the method from [8] using the following metrics:

- Measurement accuracy

We use the relative error of the measurement results as a metric to evaluate the measurement accuracy of the methods. The relative error is calculated from

$$e = \frac{|A_{meas} - \bar{A}|}{\bar{A}}, \qquad (4)$$

where A_{meas} is the average of the measurement results over an aggregation period as defined by Eq. (3) and $\bar{A}$ is the average of the real value of available bandwidth over that aggregation period.

- Measurement traffic load

We use the average number of packet fleets traversing one link to evaluate the measurement traffic load.

4.2 Simulation settings

To test whether our method works correctly as designed, we applied our method to the small network topology shown in Fig. 3 and observed the behavior in detail. We used three different types of large network topologies for comparing our method with the method from [8]: the AT&T topology [15], and generated topologies based on the Barabasi-Albert (BA) model [16], and the Waxman model [17]. We generated 10 topologies for each model using the BRITE topology generator [18]. All topologies have 523 nodes and 1304 links. We set the density of the overlay nodes to 0.2 and randomly chose the overlay nodes from among the 523 nodes. Results were averaged across 100 different choices of overlay nodes for the AT&T topology and 10 different choices for each of the BA and Waxman model topologies. For simplicity, we assume that the capacity of all IP links in the network is C and set $C = 100$ [Mbps].

We made the following assumptions about the temporal changes in the amount of traffic between overlay nodes. We assume that cross traffic occurs in some fraction α ($0 < \alpha \leqq 1$) of the paths. In the small network topology, α was set to 0.2, and in the large network topologies, it was set to 0.02. For a path O_iO_j where cross traffic occurs, let the IP links of that path be $l_1, l_2, ..., l_r$. We assume that among the paths where cross traffic occurs, the number of paths that share the link l_t ($1 \leqq t \leqq r$) is b_t. We let $b_{max} = \max\{b_1, b_2, ..., b_r\}$, and set $s_{max} = 0.9C/b_{max}$ and $s_{min} = 0.5s_{max}$. The rate of cross traffic across O_iO_j was then randomly chosen in the range $[s_{min}, s_{max}]$. Furthermore, the intervals where traffic occurs and does not occur were randomly chosen in the range [120s, 1200s].

Since we have adopted a method based on Pathload for end-to-end measurement, we select the measurement parameters by following the suggestions of the authors of Pathload [3]. In particular, we set $\tau = 12$ [s] and $\omega = 400$ [Kbps]. In the small network topology, we set the measurement duration to 400τ. In the large network topologies, the measurement duration was set to the length of 10 aggregation periods, and each aggregation period T was set to 1200τ.

4.3 Evaluation results and discussions

4.3.1 Evaluation results for the small network topology

Figure 4 shows the measurement results of paths O_1O_4 and O_3O_1 of the network in Fig. 3. The measurement results of other paths exhibited similar trends, and are thus omitted in the interest of saving space. In Fig. 4, the blue lines show the real values of available bandwidth, the pink bars show the search ranges, and the red bars show the measurement results. Because our measurement accuracy depends on the search range, we evaluate the effectiveness of our method by considering the variation of the search range. As shown in Fig. 4, the search range varies based on and tends to approach the real value of available bandwidth. When the real value changes by a large amount, the width of the search range becomes large at first but then becomes gradually smaller, and the search range quickly approaches the real value of available bandwidth. These results demonstrate that our proposed method for calculating the search range is efficient for measuring available bandwidth.

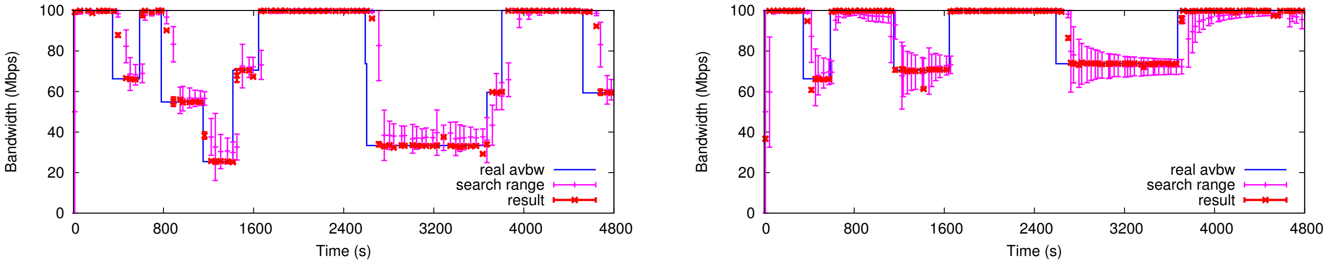
4.3.2 Evaluation results for the large network topologies

Table 1 shows the distribution of the relative errors in the measurement results for the AT&T, BA, and Waxman topologies. In particular, it shows the percentage of relative errors in the measurement results that are not smaller than 0.05, 0.1, 0.2, and 0.4. Table 2 shows the average value of the relative errors in the measurement results. The relative errors in the measurement results of our method are only approximately 65% those of the method from [8]. To explain these results, we use the evaluation results of the average number of measurements and the average number of measurement conflicts of an overlay path during an aggregation period, shown in Tabs. 3 and 4, respectively. The number of measurements in our method is much larger than that of the method in [8], but only about 12% of the measurements experience conflicts. Therefore, the measurement accuracy of our method surpasses the method from [8]. We also observe in Tabs. 1 and 2 that the Waxman topology has smaller relative error than the AT&T and BA topologies for the following reason. From the simulation results, we found that the number of half and partial overlapping paths in the Waxman topology is smaller than that in the AT&T and BA topologies. Therefore, the measurement frequency is the largest, meaning that the number of measurements is the largest, and thus the relative error is the smallest in the Waxman topology.

Table 5 shows the average number of packet fleets traversing each link per measurement, and shows that the average number of packet fleets is smaller by the proposed method compared to the method from [8]. This is because the search range in each end-to-end measurement of available bandwidth is set to $(0, C)$ in the method from [8], and so the number of packet fleets per measurement is constant across all measurements. By comparison, since the search range is calculated based on the measurement results that are exchanged between overlay nodes in the proposed method, the search ranges are narrower and closer to the real value of available bandwidth. The number of packet fleets per measurement is therefore smaller, meaning that the traffic load of each measurement is smaller in our method.

5 Conclusion

We proposed a distributed method for measuring available bandwidth in overlay networks that reduces measurement conflicts by



(a) Path O_1O_4

(b) Path O_3O_1

Fig. 4 Measurement results for the small network topology

Topology	AT&T				BA				Waxman			
Method \ Relative error	≧ 0.05	≧ 0.1	≧ 0.2	≧ 0.4	≧ 0.05	≧ 0.1	≧ 0.2	≧ 0.4	≧ 0.05	≧ 0.1	≧ 0.2	≧ 0.4
Existing method	56.600%	32.184%	9.576%	1.432%	50.994%	29.480%	9.542%	1.153%	33.411%	15.373%	3.418%	0.167%
Proposed method	41.999%	18.087%	3.260%	0.194%	35.472%	14.161%	2.546%	0.105%	26.841%	9.492%	1.385%	0.024%

Table 1 Distribution of relative errors

Method \ Topology	AT&T	BA	Waxman
Existing method	0.088	0.081	0.049
Proposed method	0.058	0.049	0.039

Table 2 Average relative errors

Method \ Topology	AT&T	BA	Waxman
Existing method	3.287	5.912	13.202
Proposed method	11.050	20.259	28.388

Table 3 Average number of measurements per aggregation period

Method \ Topology	AT&T	BA	Waxman
Existing method	0.000	0.000	0.000
Proposed method	1.554	2.177	3.379

Table 4 Average number of measurement conflicts per aggregation period

Method \ Topology	AT&T	BA	Waxman
Existing method	6.000	6.000	6.000
Proposed method	5.814	5.784	5.674

Table 5 Average number of packet fleets traversing each link per measurement

detecting overlapping paths and adjusting the measurement frequencies and measurement timings of overlay paths. We also proposed a method to improve measurement accuracy while reducing the traffic load of each measurement by exchanging measurement results among neighboring overlay nodes. Simulation results show that the relative errors in the measurement results of our method are only approximately 65% those of an existing method.

Acknowledgment

This work was supported in part by the Strategic Information and Communications R&D Promotion Programme (SCOPE) of the Ministry of Internal Affairs and Communications, Japan.

References

[1] K. Suh, C. Diot, J. Kurose, L. Massoulie, C. Neumann, D. Towsley, and M. Varvello, "Push-to-peer video-on-demand system: Design and evaluation," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 9, pp. 1706–1716, 2007.

[2] S. Liu, R. Zhang-Shen, W. Jiang, J. Rexford, and M. Chiang, "Performance bounds for peer-assisted live streaming," *SIGMETRICS Perform. Eval. Rev.*, vol. 36, pp. 313–324, June 2008.

[3] M. Jain and C. Dovrolis, "Pathload: A measurement tool for end-to-end available bandwidth," in *Proc. PAM 2002*, pp. 14–25, 2002.

[4] J. Strauss, D. Katabi, and F. Kaashoek, "A measurement study of available bandwidth estimation tools," in *Proc. ACM SIGCOMM 2003*, 2003.

[5] N. Hu and P. Steenkiste, "Exploiting internet route sharing for large scale available bandwidth estimation," in *Proc. IMC 2005*, Oct. 2005.

[6] C. Xing, M. Chen, and L. Yang, "Predicting available bandwidth of internet path with ultra metric space-based approaches," in *Proc. IEEE GLOBECOM 2009*, pp. 1–6, 2009.

[7] S. Song, P. Keleher, B. Bhattacharjee, and A. Sussman, "Decentralized, accurate, and low-cost network bandwidth prediction," in *Proc. IEEE INFOCOM 2011*, pp. 6–10, 2011.

[8] M. Fraiwan and G. Manimaran, "Scheduling algorithms for conducting conflict-free measurements in overlay networks," *Computer Networks*, vol. 52, pp. 2819–2830, 2008.

[9] Z. Qin, R. Rojas-Cessa, and N. Ansari, "Task-execution scheduling schemes for network measurement and monitoring," *Computer Communications*, vol. 33, no. 2, pp. 124 – 135, 2010.

[10] T. H. Dinh, G. Hasegawa, and M. Murata, "A low-cost, distributed and conflict-aware measurement method for overlay network services utilizing local information exchange," *IEICE Transactions on Communications*, vol. E96-B, pp. 459–469, Feb. 2013.

[11] V. J.Ribeiro, R. H. Riedi, R. G. Baraniuk, J. Navratil, and L. Cottrell, "pathchirp: Efficient available bandwidth estimation for network paths," in *Proc. PAM 2003*, 2003.

[12] N. M. M. K. Chowdhury and R. Boutaba, "A survey of network virtualization," *Computer Networks*, vol. 54, no. 5, pp. 862 – 876, 2010.

[13] J. Rubio-Loyola, A. Galis, A. Astorga, J. Serrat, L. Lefevre, A. Fischer, A. Paler, and H. Meer, "Scalable service deployment on software-defined networks," *IEEE Communications Magazine*, vol. 49, pp. 84 –93, Dec. 2011.

[14] C. L. T. Man, G. Hasegawa, and M. Murata, "Monitoring overlay path bandwidth using an inline measurement technique," *IARIA International Journal on Advances in Systems and Measurements*, vol. 1, no. 1, pp. 50–60, 2008.

[15] N. Spring, R. Mahajan, and C. Wetherall, "Measuring isp topologies with rocketfuel," in *Proc. ACM SIGCOMM 2002*, Jan. 2002.

[16] A. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, Oct. 1999.

[17] B. M. Waxman, "Routing of multipoint connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, pp. 1617–1622, Dec. 1988.

[18] BRITE: Boston university Representative Internet Topology gEnerator, available at `http://www.cs.bu.edu/brite/index.html`.