PAPER *Special Section on Quality of Communication Services Improving Quality of Life*

A low-cost, distributed and conflict-aware measurement method for overlay network services utilizing local information exchange

Tien Hoang DINH[†a)], **Go HASEGAWA**[†], ***Members***, ***and*** **Masayuki MURATA**[†], ***Fellow***

SUMMARY Measuring network resource information, including available bandwidth, propagation delay, and packet loss ratio, is an important task for efficient operation of overlay network services. Although measurement accuracy can be enhanced by frequent measurements, performing measurements with high frequency can cause measurement conflict problem that increases the network load and degrades measurement accuracy. In this paper, we propose a low-cost, distributed and conflict-aware measurement method that reduces measurement conflicts while maintaining high measurement accuracy. The main idea is that the overlay node exchanges the route information and the measurement results with its neighboring overlay nodes while decreasing the measurement frequency. This means our method trades the overhead of conducting measurements for the overhead of information exchange to enhance measurement accuracy. Simulation results show that the relative error in the measurement results of our method can be decreased by half compared with the existing method when the total measurement overheads of both methods are equal. We also confirm that exchanging measurement results contributes more to the enhancement of measurement accuracy than performing measurements.

key words: *overlay networks, network measurement, measurement conflict, distributed measurement method, information exchange*

1. Introduction

Recently, overlay networks have attracted much attention as a technology that enables early deployment of new network services without standardization processes. Applications of overlay networks include end-system multicast (e.g., Narada [1]), P2P systems (e.g., Skype [2], KaZaA [3], BitTorrent [4]), content distribution systems (e.g., Akamai [5]), and resilient routing (e.g., RON [6]).

In overlay networks, the overlay nodes are often installed on end hosts as an application program. In this case, routing and traffic control at the overlay detecting level are conducted at the end hosts, and such controls cannot be activated inside the network. On the other hand, the overlay routing inside the network becomes possible by installing overlay nodes on the routers in the network. This installation has been simplified with such techniques as network virtualization [7] and software defined network [8]. In this paper, to realize efficient routing control by overlay networks, we consider an overlay network in which the overlay nodes are deployed on the routers.

An overlay network should obtain the network resource information of the underlay network, including available bandwidth, propagation delay, and packet loss ratio, to maintain and improve the performance of network service. These metrics should be measured frequently to obtain high measurement accuracy. RON [6] is one early-stage instance that measures all paths among overlay nodes. The measurement overhead becomes $O(n^2)$, where n is the number of overlay nodes. Therefore, [9] pointed out that the number of overlay nodes that can be applied is up to around fifty. Many solutions have been proposed to reduce measurement overhead [10]–[16]. However, these methods have shortcomings in terms of measurement accuracy [10] or available measurement metrics [12], [13].

Measurement accuracy is affected not only by the way measurements are performed but also by the overlap of underlay paths among overlay nodes. Fig. 1 illustrates an example of overlapping paths. O_i and R_i ($i = 1, ..., 5$) represent overlay nodes and routers. Although paths O_1O_4 and O_2O_5 are disjointed at the overlay level, they overlap at the underlay level, i.e., they share links and routers on the path between R_1 and R_5. Therefore, the concurrent measurement tasks of paths O_1O_4 and O_2O_5 compete on the common links for network resources (e.g., processing power at routers and link bandwidth), causing high load on the common links and additional error in the measurement results.

[17] addresses this problem and proposes a method that schedules the timing of the measurement tasks of the overlay paths so that measurement conflicts can be avoided completely. However, the measurement frequency in this method is limited because of the heuristic behavior of the proposed scheduling algorithms [18]. Moreover, the methods in [10], [11], [14]–[17] require a master node to aggregate the complete topology information of the underlay (IP) network, decide measurement timings, and give instructions to each overlay node. Therefore, the amount of time and network traffic for the aggregation of topology information and instructions are large, and the performance of overlay networks decreases when changes occur in the underlay or overlay networks.

In this paper, we propose a distributed measurement method that can reduce measurement conflicts and obtain high measurement accuracy. In our proposed method, each overlay node exchanges route information with its neighboring overlay nodes to detect the overlapping paths. Overlapping paths with the same source node are measured sequentially to completely avoid measurement conflicts. Overlapping paths having different source nodes are randomly measured to reduce measurement conflicts. The overlay node

Manuscript received May 31, 2012.
Manuscript revised July 31, 2012.

[†] The authors are with the Graduate School of Information Science and Technology, Osaka University, Suita-shi, 565-0871 Japan.

a) E-mail: d-hoang@ist.osaka-u.ac.jp

DOI: 10.1587/transcom.E0.B.1

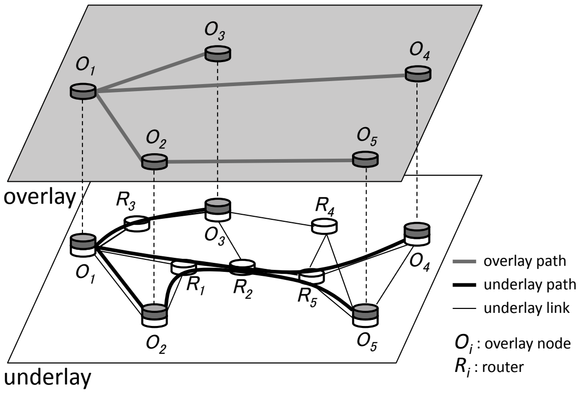


Fig. 1 Example of path overlapping

then exchanges the measurement results with its neighboring overlay nodes to statistically improve measurement accuracy. Our method can also lower the measurement frequencies to reduce overhead and measurement conflicts.

We make the following contributions in this paper:

- We propose two algorithms for detecting the overlapping paths that do not require complete topology knowledge of the IP network at each node.
- We propose a method for determining the measurement frequencies and timings of the overlapping paths to reduce measurement conflicts.
- We evaluate our method and compare it with the method in [17] by simulations with both generated and real Internet topologies.

From the simulation results, we reach the following conclusions:

- Our method detects more than 90% of the overlapping paths with less than 30% of the information exchanges of the full-mesh method.
- When the overheads of our method and the method in [17] are equal, the relative error of the measurement results of our method is less than half of the method in [17].

The remainder of this paper is organized as follows. Section 2 describes related work. In Sect. 3, we explain our method for detecting the overlapping of overlay paths. Section 4 describes our technique for reducing measurement conflicts and improving measurement accuracy. In Sect. 5, our proposed method is evaluated by simulations. We conclude this paper and discuss future work in Sect. 6.

2. Related work

RON [6] can measure many network resource information of the underlay network such as available bandwidth, propagation delay and packet loss ratio, but it suffers from a lack of scalability. Therefore, the measurement methods proposed later tried to reduce the measurement overhead from the $O(n^2)$ overhead of RON. Network tomography [10], [11], [14]–[16] is an effective approach to achieve this goal. The main idea of these methods is that they monitor only a few paths that cover all the links of the overlay network and use the measurement results of the collected paths to infer the measurement results of the remaining paths. However, the centralized behavior of these methods makes it hard for them to cope with changes or troubles that occur in the underlay network.

The measurement conflict problem, which was first addressed in [19], is considered in later work [17], [20]. The main idea of these studies is that they use heuristic algorithms from graph theory to schedule the measurement timings of paths so that the overlapping paths are measured at different timings. Although measurement conflicts can be avoided completely, the measurement frequencies are limited, so measurement accuracy is not high. We also point out that when the measurement traffic is not so intrusive, for example, when the measurement metric is latency, it is not necessary to completely avoid measurement conflicts.

Only a few measurement methods work in a distributed fashion [13], [21], and they have their own limits. The authors in [13] proposed a measurement system for available bandwidth, called ImSystemPlus, that can reduce measurement conflicts without using a master node by randomly deciding the measurement timing of overlapping paths. However, this method requires complete topology knowledge of the IP network at each overlay node. [21] proposed a measurement system in which overlay nodes estimate their virtual coordinates and exchange with each other to calculate the distances between them and infer latencies from those distances. However, this method cannot be applied to measure packet loss and bandwidth.

3. Detecting overlapping paths

3.1 Network model and definitions

We consider a network with m routers, denoted by R_i ($i = 1, ..., m$). We denote the underlay path between two routers R_i and R_j as R_iR_j. If two different paths R_iR_j and R_sR_t share at least one link, we say that R_iR_j and R_sR_t overlap with each other, or R_iR_j (R_sR_t) is an *overlapping path* of R_sR_t (R_iR_j).

Suppose that there are n ($n \leq m$) overlay nodes deployed on n routers. Density σ of the overlay nodes is defined as the ratio of the number of overlay nodes to the number of routers, i.e., $\sigma = n/m$. We denote the overlay nodes as O_i ($i = 1, ..., n$) and call the path between two overlay nodes an *overlay path*. For overlay path O_iO_j, O_i is the *source node*, and O_j is the *destination node* of the overlay path.

Figure 2 shows a classification of the overlapping state of overlay paths. In this paper, we classify overlapping states into the following three types:

- Complete overlapping: One overlay path completely includes another overlay path.
- Half overlapping: Two overlay paths share a route from the source node to a router that is not an overlay node.

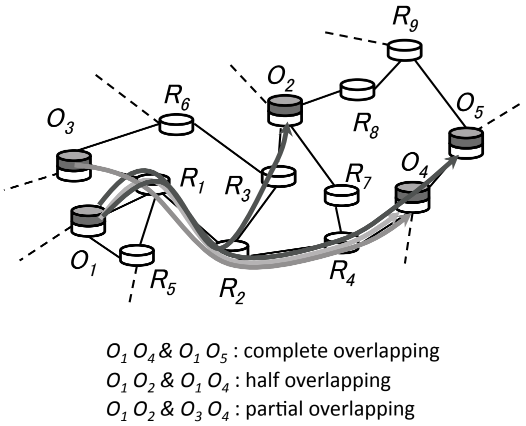

R_9
O_2
R_6
O_3
O_5
R_8
R_1
R_3
R_7
O_4
O_1
R_5
R_2
R_4
$O_1\ O_4$ & $O_1\ O_5$: complete overlapping
$O_1\ O_2$ & $O_1\ O_4$: half overlapping
$O_1\ O_2$ & $O_3\ O_4$: partial overlapping

Fig. 2 Classification of path overlapping

- Partial overlapping: Two overlay paths share a route that does not include the source node.

For example, in Fig. 2, path O_1O_4 is a complete overlapping path of O_1O_5. Paths O_1O_2 and O_1O_4 have a half overlapping relation. Path O_1O_2 is a partial overlapping path of O_3O_4.

3.2 Methods for detecting complete and half overlapping paths

Complete overlapping and half overlapping can be detected by the source node of the overlay path using `traceroute`-like tools, as described in [22]. For example, in Fig. 2, when overlay node O_1 issues `traceroute` to O_4 and O_5, complete overlapping of paths O_1O_4 and O_1O_5 can be detected. Similarly, the shared route from O_1 to router R_2 by paths O_1O_2 and O_1O_4 can be detected when O_1 issues `traceroute` to O_2 and O_4.

3.3 Method for detecting partial overlapping paths

3.3.1 Detecting algorithms

Partial overlapping cannot be precisely detected only by `traceroute`-like tools, because the source nodes of the partial overlapping paths are different. Therefore, in this subsection, we propose the following method for detecting partial overlapping paths.

We demonstrate how an overlay node O_i detects the partial overlapping paths. We denote the set of overlay paths whose source nodes are O_i, which contain at least two links and do not completely include other overlay paths as $\mathcal{S}_{O_i}$. We also denote the set of overlay paths whose destination nodes are O_i, which contain at least two links and do not completely include other overlay paths as $\mathcal{D}_{O_i}$. Note that we exclude one-link paths when defining $\mathcal{S}_{O_i}$ and $\mathcal{D}_{O_i}$ since they do not have partial overlapping paths. Also, we do not directly measure the paths that completely include other overlay paths, as described in Subsect. 4.1.1.

Algorithm 1 O_i detects the partial overlapping paths of the paths in $\mathcal{S}_{O_i}$

```
//initilization
for $O_iO_j \in \mathcal{S}_{O_i}$ do
    $C_{O_iO_j} \leftarrow \emptyset$ //set of candidates of partial overlapping paths of $O_iO_j$
    $\mathcal{N}_{O_iO_j} \leftarrow \emptyset$ //set of nodes that receives information of $O_iO_j$
end for
for $O_j \neq O_i$ do
    $\mathcal{T}_{O_i}^{O_j} \leftarrow \emptyset$ //set of paths that $O_i$ sends to $O_j$
    $\mathcal{R}_{O_i}^{O_j} \leftarrow \emptyset$ //set of paths that $O_i$ receives from $O_j$
end for

//find candidates of partial overlapping paths
for $O_iO_j \in \mathcal{S}_{O_i}$ do
    for each pair $O_iO_s$, $O_iO_t$ of half overlapping paths of $O_iO_j$ do
        if $OverlapLength(O_iO_j, O_iO_s) < OverlapLength(O_iO_j, O_iO_t)$ then
            $C_{O_iO_j} \leftarrow C_{O_iO_j} \cup \{O_sO_t\}$
        else if $OverlapLength(O_iO_j, O_iO_s) > OverlapLength(O_iO_j, O_iO_t)$ then
            $C_{O_iO_j} \leftarrow C_{O_iO_j} \cup \{O_tO_s\}$
        end if
    end for
end for

//update set of paths that $O_i$ sends to other nodes
for $O_iO_j \in \mathcal{S}_{O_i}$ do
    for $O_sO_t \in C_{O_iO_j}$ do
        $\mathcal{T}_{O_i}^{O_s} \leftarrow \mathcal{T}_{O_i}^{O_s} \cup \{O_iO_j\}$
    end for
end for

//$O_i$ exchanges information of paths with other nodes
for $O_j \neq O_i$ do
    loop
        for $O_iO_s \in \mathcal{T}_{O_i}^{O_j}$ do
            $O_i$ sends information of $O_iO_s$ to $O_j$
            $\mathcal{N}_{O_iO_s} \leftarrow \mathcal{N}_{O_iO_s} \cup \{O_j\}$
        end for
        $\mathcal{T}_{O_i}^{O_j} \leftarrow \emptyset$ //clear the set $\mathcal{T}_{O_i}^{O_j}$
        $O_i$ receives information of paths from $O_j$ and adds it to set $\mathcal{R}_{O_i}^{O_j}$
        $O_i$ detects partial overlapping between the paths in $\mathcal{S}_{O_i}$ and the paths in $\mathcal{R}_{O_i}^{O_j}$
        //update the set $\mathcal{T}_{O_i}^{O_j}$
        if there are some paths in $\mathcal{S}_{O_i}$ that overlap with at least one path in $\mathcal{R}_{O_i}^{O_j}$ and have not been sent to $O_j$ then
            Add these paths to $\mathcal{T}_{O_i}^{O_j}$
        end if
        //stop if there is no more information of paths to send
        if $\mathcal{T}_{O_i}^{O_j} = \emptyset$ then
            exit loop
        end if
    end loop
end for
```

Our method consists of two steps that detect the partial overlapping paths of each path in $\mathcal{S}_{O_i}$ and $\mathcal{D}_{O_i}$, respectively. In the first step, O_i finds the candidates of the partial overlapping paths of the paths in $\mathcal{S}_{O_i}$. O_i then exchanges the path information with the source nodes of the candidates to confirm whether they are actually partial overlapping paths. In the second step, O_i exchanges the information of the paths in $\mathcal{D}_{O_i}$ with their source nodes to detect their partial overlapping paths.

Algorithm 2 O_i detects the partial overlapping paths of the paths in $\mathcal{D}_{O_i}$

```
//$O_i$ sends path information
for $O_iO_j \in \mathcal{S}_{O_i}$ do
    $O_i$ sends information of $O_iO_j$ and $\mathcal{N}_{O_iO_j}$ to $O_j$
end for

//$O_i$ receives path information
$\mathcal{D}_{O_i} \leftarrow \emptyset$
for $O_j \neq O_i$ do
    $O_i$ receives information of $O_jO_i$ and set $\mathcal{N}_{O_jO_i}$ from $O_j$
    $\mathcal{D}_{O_i} \leftarrow \mathcal{D}_{O_i} \cup \{O_jO_i\}$
end for

//$O_i$ detects partial overlapping paths and sends to other nodes
for each pair $O_sO_i, O_tO_i \in \mathcal{D}_{O_i}$ do
    if $O_sO_i$ and $O_tO_i$ overlap with each other then
        if $O_t \notin \mathcal{N}_{O_sO_i}$ then
            $O_i$ sends information of $O_sO_i$ to $O_t$
        end if
        if $O_s \notin \mathcal{N}_{O_tO_i}$ then
            $O_i$ sends information of $O_tO_i$ to $O_s$
        end if
    end if
end for

$O_i$ receives the partial overlapping paths of paths in $\mathcal{S}_{O_i}$ from other nodes
```

Algorithm 1 shows the details of the first step. Function *OverlapLength* returns the length (number of hops) of the overlapping part between two paths. In this algorithm, O_i finds the candidates of the partial overlapping paths of each path O_iO_j in $\mathcal{S}_{O_i}$ by utilizing the information of its half overlapping paths. In detail, when O_iO_s and O_iO_t are half overlapping paths of O_iO_j and when the length of the overlapping part of O_iO_j and O_iO_s is smaller than the length of the overlapping part of O_iO_j and O_iO_t, we infer that O_sO_t is a candidate of the partial overlapping path of O_iO_j. O_i then exchanges path information with O_s to determine whether O_iO_j and O_sO_t actually have a partial overlapping relation. In this way, O_i exchanges path information with the source nodes of the candidates to decide their overlapping states. Furthermore, when receiving path information from other nodes, O_i may find new candidates of the partial overlapping paths. In that case, O_i repeats the information exchange and the decisions of the overlapping states.

We use Fig. 2 to explain how Algorithm 1 works for path O_1O_2. Set $\mathcal{S}_{O_1}$ includes O_1O_2, O_1O_3, and O_1O_4 and does not include O_1O_5 because it completely contains O_1O_4. We infer that path O_3O_4 is a partial overlapping path of O_1O_2, because the length of the overlapping part of O_1O_2 and O_1O_3 is smaller than the length of the overlapping part of O_1O_2 and O_1O_4. O_1 then exchanges path information with O_3 to confirm whether O_1O_2 and O_3O_4 actually have a partial overlapping relation.

Algorithm 2 shows the details of the second step. In this algorithm, O_i exchanges path information with other nodes to detect the partial overlapping paths of the paths in $\mathcal{D}_{O_i}$ as follows.

1. O_i receives information of each path in $\mathcal{D}_{O_i}$ from the source node (referred to as O_s) of the path.
2. O_i detects the partial overlapping paths of each path O_sO_i in $\mathcal{D}_{O_i}$ and sends information of these paths to O_s.

We also use Fig. 2 to explain how Algorithm 2 works for path O_2O_4. Set $\mathcal{D}_{O_4}$ includes O_1O_4, O_2O_4, and O_3O_4 and does not include O_5O_4 because it contains only one link. First, O_4 receives the information of paths O_1O_4, O_2O_4, and O_3O_4 from O_1, O_2, and O_3, respectively. O_4 then detects that O_1O_4, O_2O_4, and O_3O_4 are in a partial overlapping relation and sends the information of O_1O_4 and O_3O_4 to O_2.

3.3.2 Evaluation of detecting algorithms

We evaluate our proposed algorithms for detecting partial overlapping paths by simulations with two metrics, defined as follows:

- detection ratio: ratio of the number of detected partial overlapping paths to the actual number of partial overlapping paths.
- number of path information exchanges: number of times that the information of overlay path was exchanged among the overlay nodes.

Algorithm 1 includes iterations for information exchange and the decision of the overlapping states. When the number of iterations increases the detection ratio is enhanced, while the overhead of the information exchange among the overlay nodes also increases. In addition, since Algorithms 1 and 2 can be conducted independently, we set the following four detecting levels to conduct Algorithms 1 and 2 to investigate the trade-off relationships between the detection ratio and the information exchange overhead.

- detecting level 1: run Algorithm 1 with one iteration.
- detecting level 2: run Algorithm 1 with two iterations.
- detecting level 3: run Algorithm 1 completely.
- detecting level 4: run Algorithms 1 and 2 completely.

For the underlay network topology, we used the AT&T topology obtained from [23]. We also utilized generated topologies based on BA [24] and random models [25]. We generated ten topologies for each model using the BRITE topology generator [26]. All topologies have 523 nodes and 1304 links. We set the density of the overlay nodes to 0.2 and randomly chose them. For averaging the results, the

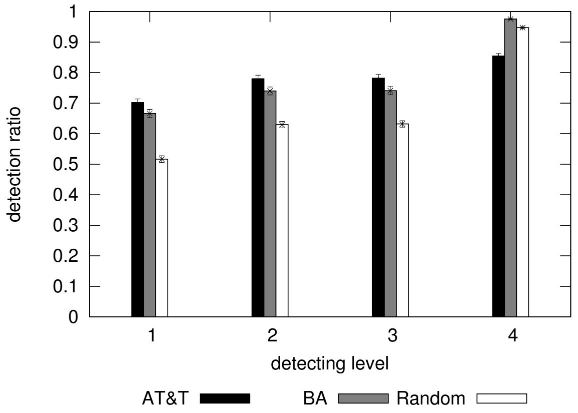


Fig. 3 Average detection ratio of partial overlapping paths

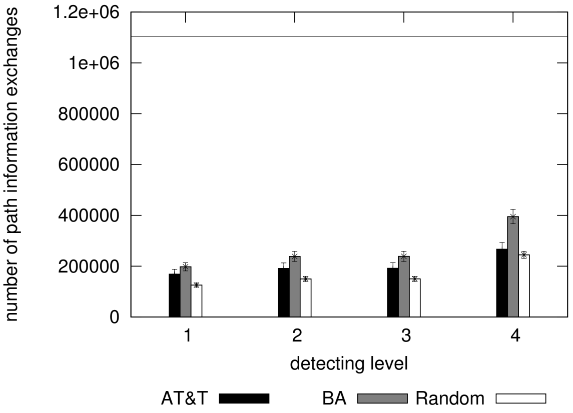


Fig. 4 Average number of path information exchanges

choice of the overlay nodes was taken 100 times for the AT&T topology and ten times for each topology of the BA and random models.

We compared our method with the full-mesh method when evaluating the number of path information exchanges. In the full-mesh method, each overlay node sends information of all overlay paths departing from it to all other overlay nodes. When the number of overlay nodes is n, the number of path information exchanges of the full-mesh method is $n(n-1)^2$, which becomes 1,103,336 in the evaluation results.

Figures 3 and 4 show the average values and 95% confidence intervals of detection ratio of the partial overlapping paths and the number of path information exchanges, respectively. The black, gray and white bars show the results of the AT&T topology, the BA topologies, and random topologies, respectively. The line in Fig. 4 represents the number of path information exchanges of the full-mesh method. As shown in these figures, our method needs only 1/6 and 1/3 of the path information exchanges to detect about 60% and 90% of the partial overlapping paths at detecting levels 1 and 4, respectively. The results of detecting levels 2 and 3 are very close, meaning that we only need to run two iterations of the exchange loop of Algorithm 1.

4. Measurement method for overlay paths

In this section, we propose a method for reducing the measurement conflicts based on the status of the path overlapping detected by the method in Sect. 3. We explain the proposed method by describing the detailed behavior for an overlay path O_iO_j. First, node O_i detects the overlapping paths of path O_iO_j with the method described in Sect. 3. If path O_iO_j has no overlapping paths, it is unnecessary to consider a method for reducing measurement conflicts. Therefore, we are only concerned with the case when path O_iO_j overlaps with other overlay paths.

We consider the following two cases of overlapping states:

1. When path O_iO_j completely includes other overlay paths, overlay path O_iO_j is not measured directly.
2. When path O_iO_j does not include other overlay paths, we adjust the frequency and timing of the measurements to reduce the measurement conflicts.

The detailed mechanisms for the above two cases are described in Subsects. 4.1.1 and 4.1.2, respectively. In Subsect. 4.2, we propose a statistical method for improving the accuracy of the measurement results.

Finally, in Subsect. 4.3, we describe the entire procedure for each overlay node to measure the overlay paths departing from it.

4.1 Reducing measurement conflicts

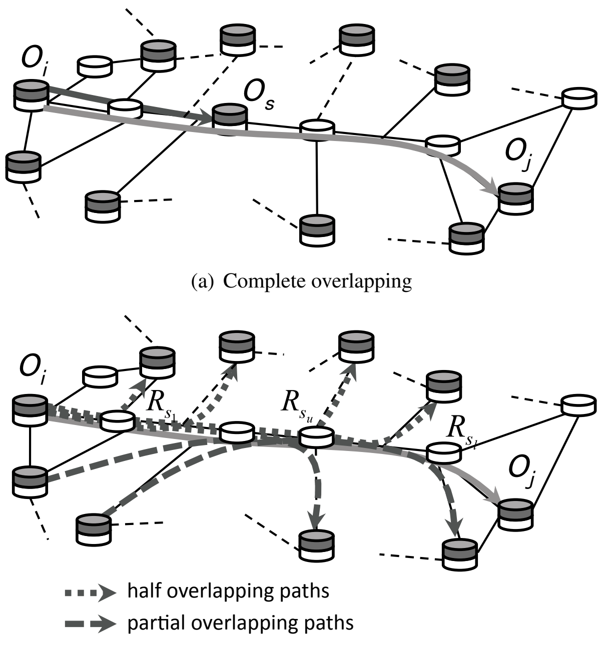
4.1.1 Complete overlapping

In this case, the overlay path that includes the other overlay paths is not measured directly. Instead, the measurement result is estimated based on the measurement results of the overlay paths included in it.

We use Fig. 5(a) to explain this method. As shown in Fig. 5(a), path O_iO_j completely includes path O_iO_s. When O_i issues `traceroute` to O_j, the `traceroute` packet goes through O_s, which learns that it is on path O_iO_j. O_s then measures path O_sO_j and transmits the result to O_i, which also learns that O_s is on path O_iO_j, based on the `traceroute` result. Then O_i does not directly measure path O_iO_j; it only measures path O_iO_s. O_i estimates the measurement result of path O_iO_j from the measurement result of path O_iO_s and that of path O_sO_j received from O_s. See [22] for details. Note that this method dramatically reduces the number of measurement paths, especially when the density of the overlay nodes is large [22]. Furthermore, the reasonable measurement accuracy of such a spatial composition method has been confirmed [27].

4.1.2 Half and partial overlapping

Here, we assume that O_iO_j has $(G_{i,j}-1)$ half overlapping paths $(G_{i,j} \geq 1)$, as shown in Fig. 5(b). For simplicity,



(a) Complete overlapping

(b) Half and partial overlapping

Fig. 5 Examples for explaining the proposed measurement method

we rewrite $G_{i,j}$ as G. We denote path O_iO_j as path 1, and each of its half overlapping paths as path p ($2 \leq p \leq G$). Furthermore, we assume that, with the method described in Sect. 3 to detect partial overlapping paths, path p ($1 \leq p \leq G$) has $(K_p - 1)$ partial overlapping paths ($K_p \geq 1$).

Overlay node O_i can avoid the measurement conflicts between half overlapping paths 1, 2, ... and G simply by measuring them sequentially. On the other hand, because the source nodes of the partial overlapping paths of path p are different, measurement conflicts between them cannot be avoided completely. Therefore, we propose a technique that combines a sequential measurement for half overlapping paths and a random measurement for partial overlapping paths.

We define the *measurement frequency* as follows. We assume that the time required for each measurement task is identical for all overlay paths and denote it as τ. We also assume that the measurement results of path p are aggregated in the time duration of T_p ($T_p \geq \tau$). We call T_p an *aggregation period*. When a path is measured q ($q \leq T_p/\tau$) times at an aggregation period, its measurement frequency at that aggregation period is defined as $f_p = q\tau/T_p$.

We introduce β_p as a value that reflects the dispersion of the measurement results of path p at an aggregation period. Note that the method to determine β_p is beyond the scope of this paper. β_p can be calculated based on the statistics of the measurement results or using the method in [13]. We set measurement frequency f_p proportional to β_p for all paths, i.e., $f_1/\beta_1 = f_2/\beta_2 = ... = f_G/\beta_G$. To avoid measurement conflicts between half overlapping paths, the sum of their measurement frequencies should be equal to or less than one, i.e., $\sum_{p=1}^{G} f_p \leq 1$. So we have $f_p \leq \beta_p/(\sum_{s=1}^{G} \beta_s)$.

To reduce the probability of measurement conflicts between path p and its $(K_p - 1)$ partial overlapping paths, we set the measurement frequency of path p to a value equal to or less than $1/K_p$, i.e., $f_p \leq 1/K_p$. In addition, we keep the measurement frequencies as large as possible to obtain as many measurement results as possible. Therefor, the measurement frequency of path p is decided based on the following equation:

$$f_p \quad = \quad \min\{\beta_p/(\sum_{s=1}^{G} \beta_s), 1/K_p\}. \qquad (1)$$

Next, we explain our method for randomly deciding the measurement timings of path p so that the probability that the measurement of path p is carried out becomes f_p. We define a *measurement cycle* for the measurements of paths 1, 2, ... and G. We also divide the measurement cycle into multiple *measurement time slots*, each of which is assigned to the measurement of each path. We consider a scheme for allocating the measurement timings of paths p to these measurement time slots as follows.

When a path is measured at one measurement time slot of the measurement cycle, the probability that the measurement of the path is carried out becomes $1/G$. Therefore, we compare f_p with $1/G$ when considering the measurement timings of path p. We assume that $f_1 \geq f_2 \geq ... \geq f_G$ without loss of generality. For convenience, we define dummy value $f_0 = 1$. Since $\sum_{s=1}^{G} f_s \leq 1$, $0 \leq l < G$ exists, such that $f_0 \geq ... \geq f_l \geq 1/G \geq f_{l+1} \geq ... \geq f_G$.

If $l = 0$, meaning $f_p \leq 1/G, \forall 1 \leq p \leq G$, one measurement time slot in the measurement cycle is enough to allocate measurement timings for each path p.

On the other hand, $l > 0$ means that for path s where $s > l$, one measurement time slot is enough to allocate its measurement timings. For path t where $t \leq l$, one measurement time slot is not enough for allocating its measurement timings to satisfy its measurement frequency. In this case, the measurement time slot allocated to path s where $s > l$ is also used to measure path t where $t \leq l$ when path s is not measured.

In detail, we propose the following scheme for allocating the measurement timings of all paths.

1. Randomly decide the measurement order of path p ($1 \leq p \leq G$) at one measurement circle, and allocate the measurement time slot for each path.
2. - If $l = 0$,
     We measure path p with the probability of Gf_p at the measurement time slot allocated to it.
   - If $l \geq 1$,
     - For path t where $t \leq l$, we measure it at the measurement time slot allocated to it.
     - For path s where $s > l$, we measure it with the probability of Gf_s at the measurement time slot allocated to it.

If path s ($s > l$) is not measured, the measurement time slot is used to measure path t ($t \leq l$) with the probability of $(f_t - 1/G)/\delta$, where $\delta = \sum_{s=l+1}^{G} (1/G - f_s)$.

4.2 Statistical method for improving the accuracy for measurement results

In the proposed measurement methods in Subsect. 4.1, because it is impossible to completely avoid measurement conflicts with partial overlapping paths, the accuracy of the measurement results decreases due to measurement conflicts. Therefore, in our proposed method, overlay nodes exchange measurement results and use statistical processing to improve measurement accuracy. We assume the measuring metric is delay.

We use Fig. 5(b) to explain the method for path O_iO_j. We assume that the overlapping parts of O_iO_j and its half and partial overlapping paths are divided by routers R_{s_1}, R_{s_2}, ..., R_{s_l}. In the proposed method, the delay measurements are individually conducted for overlapping parts $R_{s_1}R_{s_2}$, $R_{s_2}R_{s_3}$, ..., $R_{s_{l-1}}R_{s_l}$ as well as for end-to-end path O_iO_j. In detail, O_i measures the delays to routers R_{s_1}, R_{s_2}, ..., R_{s_l} and calculates the delay of $O_iR_{s_1}$, $R_{s_1}R_{s_2}$, ..., $R_{s_{l-1}}R_{s_l}$ and $R_{s_l}O_j$ as follows, where the delays of $O_iR_{s_1}$, $O_iR_{s_2}$, ..., $O_iR_{s_l}$, and O_iO_j are denoted as $t_{O_iR_{s_1}}$,$t_{O_iR_{s_2}}$,...,$t_{O_iR_{s_l}}$,$t_{O_iO_j}$, respectively.

$$\begin{aligned} t_{R_{s_k}R_{s_{k+1}}} &= t_{O_iR_{s_{k+1}}} - t_{O_iR_{s_k}} \quad ,k = 1,...,l-1 \\ t_{R_{s_l}O_j} &= t_{O_iO_j} - t_{O_iR_{s_l}} \end{aligned}$$

When part $O_iR_{s_1}$ or $R_{s_k}R_{s_{k+1}}$ is the overlapping part of O_iO_j and its half overlapping path O_iO_s, $t_{O_iR_{s_1}}$ or $t_{R_{s_k}R_{s_{k+1}}}$ is used to calculate the measurement results of both paths O_iO_j and O_iO_s. When part $R_{s_k}R_{s_{k+1}}$ or $R_{s_l}O_j$ is the overlapping part of O_iO_j and its partial overlapping path O_uO_v, O_i sends $t_{R_{s_k}R_{s_{k+1}}}$ or $t_{R_{s_l}O_j}$ and its measurement timing to O_u, so that O_u can use $t_{R_{s_k}R_{s_{k+1}}}$ or $t_{R_{s_l}O_j}$ to calculate the measurement result of path O_uO_v.

Finally, we use statistical processing for the data obtained by information exchange to calculate the measurement result of path O_iO_j. First, using the gathered values with the above method, we obtain the average value of the measurement results of $O_iR_{s_1}$, $R_{s_1}R_{s_2}$, ..., $R_{s_{l-1}}R_{s_l}$, and $R_{s_l}O_j$, which are denoted as $\bar{t}_{O_iR_{s_1}}$, $\bar{t}_{R_{s_1}R_{s_2}}$, ..., $\bar{t}_{R_{s_{l-1}}R_{s_l}}$, and $\bar{t}_{R_{s_l}O_j}$, respectively. The measurement result of path O_iO_j is then calculated as follows.

$$\bar{t}_{O_iO_j} = \bar{t}_{O_iR_{s_1}} + \sum_{k=1}^{l-1} \bar{t}_{R_{s_k}R_{s_{k+1}}} + \bar{t}_{R_{s_l}O_j} \qquad (2)$$

The main idea of the above method is that source nodes of partial overlapping paths exchange measurement results of the overlapping parts to improve the measurement accuracy of these parts, and consequently improve the measurement accuracy of the whole path. Therefore, this method can be applied similarly to the metrics that the measurement results of overlapping parts can be obtained from the measurement results of the paths from the source node to the routers in the overlapping parts. These metrics include latency, loss rate, jitter, etc.

However, when the metric is bandwidth-related information such as available bandwidth or throughput, because the measurement results of overlapping parts can not be obtained, we can not apply the above method. The methods for bandwidth-related metrics are our future work.

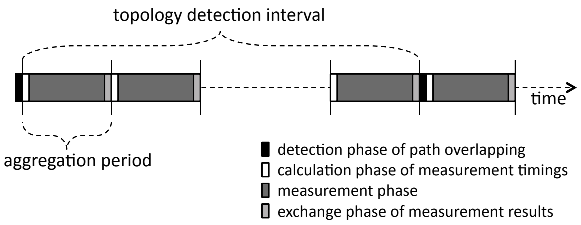

topology detection interval
time
aggregation period
▮ detection phase of path overlapping
▯ calculation phase of measurement timings
▮ measurement phase
▯ exchange phase of measurement results

Fig. 6 Measurement procedure

4.3 Measurement procedure

The measurement procedure of an overlay node includes the following four phases:

- Detection phase of path overlapping
  The overlay nodes detect the path overlapping using the method described in Sect. 3.
- Calculation phase of measurement timings
  The measurement frequencies and timings are calculated based on the status of the path overlapping, as described in Subsect. 4.1.
- Measurement phase
  The measurements are performed at the calculated measurement timings.
- Exchange phase of measurement results
  The overlay nodes exchange measurement results and calculate the measurement results of the overlay paths, as described in Subsect. 4.2.

Figure 6 illustrates the relationships among phases. The phases of the calculations of measurement timings, the measuring, and the measurement results exchange are performed at each aggregation period. Because the frequency of the change in the underlay network is generally smaller than the frequency of the change in the measurement results, the interval between two phases of path overlapping detection is larger than an aggregation period. We call this interval a *topology detection interval*. In general cases, the length of detection phase of path overlapping is much smaller than that of measurement phase, because in detection phase of path overlapping, the actions of detecting and exchanging path information are performed immediately with no waiting time, while in measurement phase, measurements are performed several times, and there are large intervals between measurements to reduce measurement conflicts. The overheads of these phases are evaluated and discussed in

Subsect. 5.2.2.

5. Performance evaluation

In this section, we evaluate the performance of our proposed method by simulation experiments. We explain the evaluation method in Subsect. 5.1 and present evaluation results and discussions in Subsect. 5.2.

5.1 Evaluation method

We compared the proposed method with an existing method [17], which we briefly explain and make some assumptions about for comparison. We then explain the evaluation metrics and the simulation settings.

5.1.1 Existing method [17]

In the method in [17], a measurement task on an overlay path is represented by a vertex in a graph. Two vertexes that represent the measurement tasks on overlapping paths are connected by an edge. The authors proposed some heuristic algorithms from graph theory to divide the vertexes into some groups, so that each group contains only disconnected vertexes which represent measurement tasks of non-overlapping paths. The measurement tasks represented by vertexes in the same group are simultaneously performed, while the measurement tasks represented by vertexes in the different groups are sequentially performed. Therefore, measurement conflicts between overlapping paths are avoided completely.

However, in [17], a detail measurement method for applying these algorithms is not mentioned. Therefore, to compare it with our method, we assume that the method in [17] is applied to a centralized measurement system like the one described in [11]. In this system, a *master node* aggregates the information of overlay paths from other overlay nodes, schedules measurement timings for the overlay paths using the method in [17], instructs other overlay nodes to measure, and aggregates the measurement results from the other overlay nodes.

5.1.2 Evaluation metrics

Here, we assume the measuring metric is delay. We compare the proposed method and the method in [17] with 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.
- System overhead
  We consider the following three kinds of overheads in conducting the measurements.
  - Path information accessing overhead
    This is caused when each overlay node uses `traceroute`-like tools to access the information of the overlay paths.
  - Measurement overhead
    This is caused when performing measurements on the overlay paths.
  - Information exchange overhead
    This is caused when overlay nodes exchange information of overlay paths and measurement results with other overlay nodes.

The relative error of the measurement result is calculated by:

$$\epsilon = \frac{|\bar{t} - t^*|}{t^*} \tag{3}$$

where t^* and $\bar{t}$ are the real delay and average values of the measurement results, respectively.

We use the M/M/1 queueing model for each link in the network to calculate t^* and $\bar{t}$. We assume that each measurement on a link causes the increase in the link utilization, that results in the increase of the delay and delay jitter at the link. When the number of concurrent measurements on a link increases, the link utilization also greatly increases, causing additional error in the delay measurements.

The system overhead, denoted by A, is calculated by:

$$A = \frac{s_a + s_m + s_e}{d} \tag{4}$$

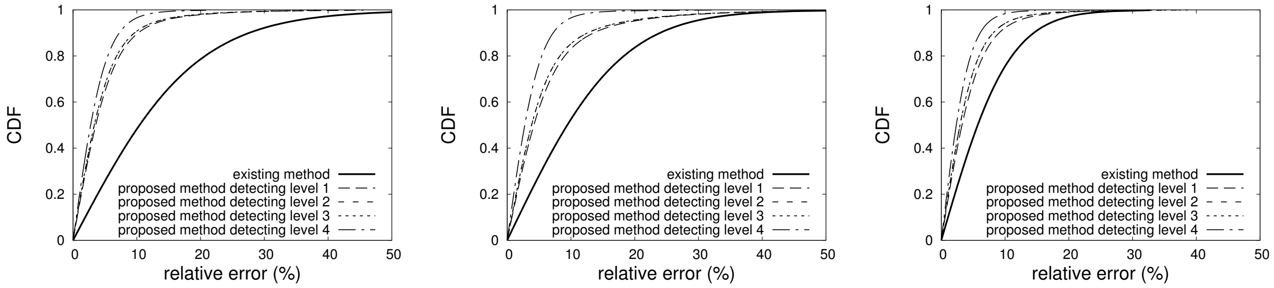
where d is the duration during which the measurements were performed, and s_a, s_m and s_e are the sizes of the data packets used for accessing the path information, measuring, and exchanging the path information and the measurement results, respectively. We use second as the unit of d and bit as the unit of s_a, s_m and s_e. Therefore, the unit of A is bit per second (bps).

5.1.3 Simulation settings

In obtaining the following simulation results, our assumptions on the network topologies, the number and the distribution of overlay nodes are the same as those mentioned in Subsect. 3.3.2.

Value β_p, which is used for calculating the measurement frequencies by Eq. (1), is determined based on the coefficient of variance of the measurement results. Furthermore, we adjust the measurement frequencies in our method so that the system overheads of the proposed method and the method in [17] are the same.

We assume that we utilize `traceroute` to access information of overlay paths, and use `ping` to measure their delays. The size of each `traceroute` packet and `ping` packet is 28 and 475 bytes, respectively. We set the time of each measurement task $\tau = 1$ (second). An aggregation period is set to one hour, and an topology detection interval is set to ten hours. We set the utilization of each link in the network to 0.5 and assume that each measurement task increases the link utilization by 0.005.



(a) AT&T topology (b) BA topology (c) Random topology

Fig. 7 Relative error of measurement results

Method Topology	existing method	proposed method detecting level 1	proposed method detecting level 2	proposed method detecting level 3	proposed method detecting level 4
AT&T	10.626	7.918	8.213	8.211	6.798
BA	16.034	10.531	10.593	10.602	9.286
Random	37.753	20.424	20.538	20.538	19.162

Table 1 Average number of measurements during an aggregation period

Method Topology	existing method	proposed method detecting level 1	proposed method detecting level 2	proposed method detecting level 3	proposed method detecting level 4
AT&T	10.626	130.323	136.889	136.852	168.294
BA	16.034	141.577	148.918	149.077	210.704
Random	37.753	187.547	203.068	203.199	277.161

Table 2 Average number of measurement results received during an aggregation period

Method Topology	existing method	proposed method detecting level 1	proposed method detecting level 2	proposed method detecting level 3	proposed method detecting level 4
AT&T	1.000	1.031	1.029	1.029	1.022
BA	1.000	1.030	1.027	1.027	1.018
Random	1.000	1.040	1.036	1.036	1.022

Table 3 Average number of concurrent measurements of one link

5.2 Evaluation results and discussions

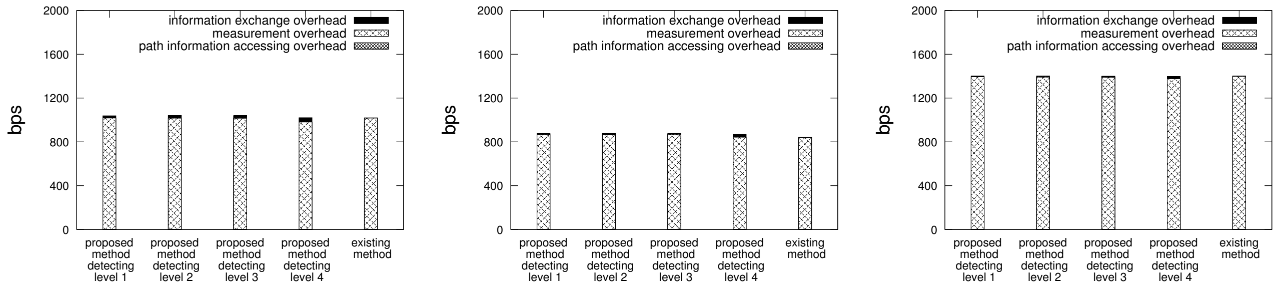
5.2.1 Measurement accuracy

Figure 7 shows the distribution of the relative error in the measurement results. The relative errors in our method are about half of those in the method in [17]. In our method, the relative errors decrease from detecting levels one to four, and the measurement accuracy of detecting level four greatly surpasses the other detecting levels.

To explain these results, we use the evaluation results of the parameters related to measurement accuracy. Tables 1 and 2 show the average number of measurements of an overlay path and the average number of the measurement results of a link (in our method) or a path (in the method in [17]) gathered during an aggregation period, respectively. In our method, as explained in Subsect. 4.2, the aggregated measurement results of each link of an overlay path include the results obtained from the measurements performed by its source node and the results received from other overlay nodes. On the other hand, in the method in [17], because measurement results are not exchanged among overlay nodes, the number of aggregated measurement results of an overlay path equals its measurement times. Table 3 shows the average number of concurrent measurements performed at a link. In the method in [17], because the measurement conflicts are avoided completely, this value remains one for all links. In our method, although the measurement conflicts cannot be avoided completely, we reduce them by adjusting the measurement frequencies based on the status of the path overlapping. Therefore, the average number of concurrent measurements of a link is very close to one.

As shown in these tables, in our method, although the number of measuring times is smaller than that in the method in [17], the number of aggregated measurement results is much larger, while the number of measurement conflicts is small. Therefore, the measurement accuracy of our method surpasses the method in [17].

We also observe in Tables 2 and 3 that when the detecting level of the proposed method is four, the number of measurement results is the largest, whereas the number of concurrent measurements is the smallest. This results in



(a) AT&T topology (b) BA topology (c) Random topology

Fig. 8 Average system overhead of one link

that the measurement accuracy at detecting level four is better than those at other detecting levels.

5.2.2 System overhead

Figure 8 shows the average values of the system overhead of the method in [17] and our proposed method with four detecting levels. The system overheads of these methods are almost equal. Furthermore, the measurement overhead occupies the most part of the system overhead, and the information exchange overhead is very small while the path information accessing overhead is negligible. This is because of the following two reasons. First, the size of the measurement traffic is much larger than the size of the traffic of information exchange and path information accessing. Second, the access frequency of path information is smaller than the measurement frequency, because the frequency of the change in the underlay network is generally smaller than the frequency of the change in the measurement results. In our method, the information exchange overhead of detecting level four is slightly larger while the measurement overhead is smaller than those of the other detecting levels. This means that by shifting some amount of overhead from measurement to information exchange, we can significantly improve the measurement accuracy.

We also observe in Figs. 7 and 8 that random topology has the smallest relative error but the largest system overhead compared with AT&T and BA topologies. We explain these results as follows. From the simulation results, we found that the number of half overlapping paths and partial overlapping paths in random topology is smaller than that in AT&T and BA topologies. Therefore, in the method in [17], the number of overlay paths that can be measured concurrently is the largest, meaning that the measurement frequency and the measurement overhead are the largest in random topology. Because the system overhead is occupied mostly by the measurement overhead, the system overhead is also the largest in random topology. Furthermore, because the measurement frequency in random topology is the largest among three network topologies, the relative error becomes the smallest. In our method, because we adjust measurement frequency of our method so that the method in [17] and our method have the same system overhead, we have the same result with the method in [17].

Figure 9 shows the distribution of system overhead on the links in the network. In the method in [17], the overhead is concentrated at several links, while in our method, the overhead is better balanced between links. This is one of side-effects of our hop-by-hop delay measurement method explained in Subsect. 4.2.

We finally conclude that from the results in Figs. 7, 8 and 9, in our method, the detecting level four is the most effective for improving measurement accuracy. Note that the detecting levels one and two are still useful, because of the following two reasons. First, although measurement accuracy in detecting level one or two is slightly worse than that in the detecting level four, it is still much better than that of the method in [17]. Second, it is easier to implement the proposed method at detecting level one or two since we only need to run Algorithm 1 with one or two iterations.

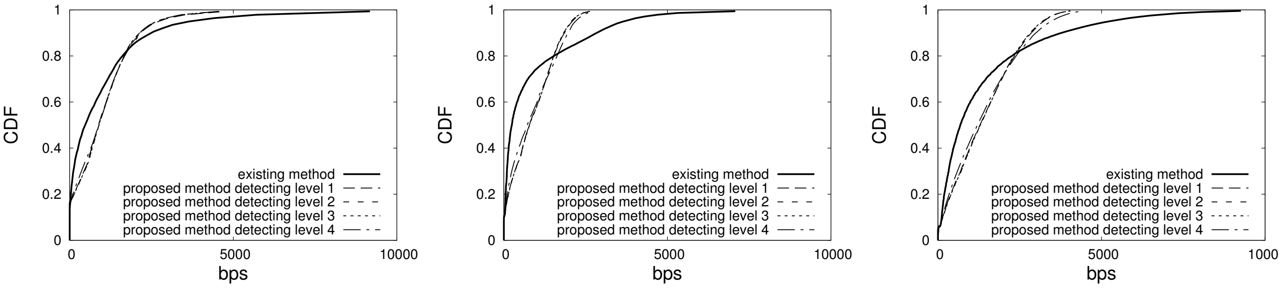
6. Conclusion

In this paper, we proposed a distributed overlay network measurement method that reduces the measurement conflicts by detecting the path overlapping and adjusting the measurement frequencies and the measurement timings of overlay paths. We also proposed a method to improve measurement accuracy by exchanging measurement results among neighboring overlay nodes. Simulation results show that the relative error in the measurement results of our method can be decreased by half compared with the existing method when the total overheads of both methods are equal. We also confirmed that exchanging measurement results contributes more to the enhancement of measurement accuracy than performing measurements.

In the future, we plan to construct a measurement system that applies the proposed method and investigate its effectiveness in real environments.

Acknowledgment

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



(a) AT&T topology (b) BA topology (c) Random topology

Fig. 9 Distribution of system overhead of all links in network

References

[1] Y. Chu, S. Rao, S. Seshan, and H. Zhang, "A case for end system multicast," IEEE Journal on Selected Areas in Communications, vol.20, no.8, pp.1456 – 1471, Oct. 2002.

[2] Skype Home Page, available at `http://www.skype.com`.

[3] KaZaA Home Page, available at `http://www.kazaa.com`.

[4] BitTorrent Home Page, available at `http://www.bittorrent.com`.

[5] Akamai Home Page, available at `http://www.akamai.com`.

[6] D. Andersen, H. Balakrishnan, M. Kaashoek, and R. Morris, "Resilient overlay networks," Proc. SOSP 2001, Oct. 2001.

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

[8] 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, no.12, pp.84 –93, Dec. 2011.

[9] A. Nakao, L. Peterson, and A. Bavier, "Scalable routing overlay networks," ACM SIGOPS Operating Systems Review, vol.40, pp.49–61, Jan. 2006.

[10] C. Tang and P. McKinley, "On the cost-quality tradeoff in topology-aware overlay path probing," Proc. ICNP 2003, Nov. 2003.

[11] Y. Chen, D. Bindel, H. Song, and R. Katz, "An algebraic approach to practical and scalable overlay network monitoring," Proc. ACM SIGCOMM 2004, Aug. 2004.

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

[13] 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.

[14] Y. Gu, G. Jiang, V. Singh, and Y. Zhang, "Optimal probing for unicast network delay tomography," Proc. IEEE INFOCOM 2010, March 2010.

[15] D. Ghita, H. Nguyen, M. Kurant, K. Argyraki, and P. Thiran, "Netscope: Practical network loss tomography," Proc. IEEE INFOCOM 2010, March 2010.

[16] A. Krishnamurthy and A. Singh, "Robust multi-source network tomography using selective probes," Proc. IEEE INFOCOM 2012, March 2012.

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

[18] D.T. Hoang, G. Hasegawa, and M. Murata, "A distributed measurement method for reducing measurement conflict frequency in overlay networks," Proc. IEEE CQR 2011, pp.1–6, May 2011.

[19] P. Calyam, C. Lee, E. Ekici, M. Haffner, and N. Howes, "Orchestration of network-wide active measurements for supporting distributed computing applications," IEEE Transactions on Computers, vol.56, no.12, pp.1629 –1642, Dec. 2007.

[20] 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.

[21] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: A decentralized network coordinate system," Proc. the ACM SIGCOMM 2004, pp.15–26, Aug. 2004.

[22] G. Hasegawa and M. Murata, "Scalable and density-aware measurement strategies for overlay networks," Proc. ICIMP 2009, pp.21–26, May 2009.

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

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

[25] B.M. Waxman, "Routing of multipoint connections," IEEE Journal on Selected Areas in Communications, vol.6, no.9, pp.1617–1622, Dec. 1988.

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

[27] G. Hasegawa and M. Murata, "Accuracy evaluation of spatial composition of measurement results in overlay networks (in Japanese)," IEICE technical report (NS2010-16), vol.110, no.39, pp.1–6, May 2010.

Tien Hoang Dinh received the M.E. degree in Information Science and Technology from Kyoto University, Japan, in 2007. He is now a D.E. candidate at Graduate School of Information Science and Technology, Osaka University. His research interest includes overlay networks and network measurement.

Go Hasegawa received the M.E. and D.E. degrees in Information and Computer Sciences from Osaka University, Japan, in 1997 and 2000, respectively. From July 1997 to June 2000, he was a Research Assistant of Graduate School of Economics, Osaka University. He is now an Associate Professor of Cybermedia Center, Osaka University. His research work is in the area of transport architecture for future highspeed networks and overlay networks. He is a member of the IEEE.

Masayuki Murata received the M.E. and D.E. degrees in Information and Computer Science from Osaka University, Japan, in 1984 and 1988, respectively. In April 1984, he joined Tokyo Research Laboratory, IBM Japan, as a Researcher. From September 1987 to January 1989, he was an Assistant Professor with Computation Center, Osaka University. In February 1989, he moved to the Department of Information and Computer Sciences, Faculty of Engineering Science, Osaka University. In April 1999, he became a Professor of Cybermedia Center, Osaka University, and is now with Graduate School of Information Science and Technology, Osaka University since April 2004. He has more than five hundred papers of international and domestic journals and conferences. His research interests include computer communication network architecture, performance modeling and evaluation. He is a member of IEEE, ACM and IEICE. He is a chair of IEEE COMSOC Japan Chapter since 2009. Also, he is now partly working at NICT (National Institute of Information and Communications Technology) as Deputy of New-Generation Network R&D Strategic Headquarters.