

# RichMap: Combining the Techniques of Bandwidth Estimation and Topology Discovery

Hui Zhou and Yongji Wang

**Abstract**—The ability to characterize the links on network topology is of great importance in both research and practice. Existing approaches capture only the topology, regardless of many critical link properties such as bandwidth and utilization. We present RichMap, an active probing system, to measure all link available bandwidth on network topology from an arbitrary end node. To do this, RichMap not only needs to discover the complete topology of surrounding networks without any prerequisite, but it also has to capture link available bandwidth by measuring paths in the topology. The accuracy and efficiency of RichMap have been verified through both simulations and network experiments over 12 diverse networks. We analyzed the result of experiments, and found that RichMap is able to accurately and efficiently capture all link available bandwidth when most of the nodes are no more than six links away from the source node.

**Index Terms**—Available bandwidth, end-to-end path, Internet topology, packet dispersion, traceroute.

## I. INTRODUCTION

Measuring link available bandwidth (available-bw) on network topology is of great importance in both research and practice. Researchers need to understand the macroscopic properties of the Internet topology in a rich context. In addition, both Internet Service Providers (ISPs) and network operators need measurement facilities to monitor their administrative networks, as well as to detect the congested and underutilized links. Furthermore, various bandwidth-sensitive applications, e.g. multimedia streaming and peer-to-peer network conference, require the information of available-bw to adjust their transmission strategies in a timely manner.

The network topology, in this paper, is a graph whose nodes represent either routers or end hosts and whose links represent adjacencies between nodes [13]. Here, two nodes are adjacent if one is exactly an IP-level hop away from the other. Link available-bw is the maximum throughput that a link can provide to a data flow without affecting other traffic in the link [16]. In the literature, a related notation that has been widely studied is the available-bw of end-to-end path. A path typically starts from a source node, crosses a sequence of intermediate links, and finally ends at a destination node. Specifically, the path available-bw is just the available-bw of bottleneck link that limits the end-to-end throughput [14],

and the source node is the location where we deploy measurement software.

In the last fifteen years, a number of approaches have been introduced to automatically discover the network topology. These approaches typically apply traceroute [18] probes to infer the router adjacencies. A traceroute probe consists of an IP packet whose Time-To-Live (TTL) field is set to a designated value (e.g.  $x$ ); after traveling across  $(x-1)$  routers, the probe triggers an ICMP response from the  $x$ th router [25]. The source node receives the ICMP packet, and extracts the IP address of the  $x$ th router out from this ICMP packet. By sending traceroute probes with TTL set to different values, the source node can obtain the IP addresses of all routers along the path. A path is built after its nodes and links are all detected, and the topology is gradually constructed with more and more paths.

Along a different research thread, many other approaches have been proposed to capture the path available-bw by actively probing the targeted path with packet trains, i.e. series of packets. The underlying assumption is that the dispersion of a long packet train is inversely proportional to the path available-bw. If the available-bw is higher than the probing data-rate, the transmission rate of the packet train will remain unchanged as it travels across all links to the destination node. Or else the dispersion must be expanded by at least one link. Therefore, the path available-bw can be obtained through monitoring the dispersion of packet trains sent at various data-rates [15], [33].

However, measuring link available-bw on the topology has received little attention, and it has been considered as a challenging task. First, the design of the network infrastructure does not provide explicit support for an end node to capture the information of network internals. Second, both the available-bw and the topology are hard to characterize because they exhibit high dynamics in a broad range of timescales [16], [3]. Third, there are not efficient methods to validate either the accuracy of available-bw or the completeness of topology.

In addition, most of the above approaches require software deployed in more than one node. For example, to ensure the accuracy of available-bw, some approaches deploy software in both the source and destination nodes of targeted paths [16]. Another example is that some approaches will map the Internet by collecting topology information from a set of source nodes that work in different positions [8]. But methods applied at a single node are generally more flexible than those demand more than one node. The reason is that one can deploy new software at his own network node, but rarely at other nodes that are not under his administrative control.

Therefore, we focus on utilizing a single and arbitrary end node. To make it possible that one can measure link

Manuscript received November 16, 2006; revised April 6, 2007. This work is supported by the National Natural Science Foundation of China grants (60673022, 60273026, and 60473060), the Chinese National "863" High-Tech Program (2005AA113140), and the Microsoft Fellowship 2005.

Hui Zhou is with the Institute of Software, Chinese Academy of Sciences, Beijing, China (phone: 86-10-62581294; fax: 86-10-62581294; e-mail: hzhou@itechs.iscas.ac.cn). Yongji Wang is with the National Key Laboratory for Computer Science, Beijing, China (e-mail: ywang@itechs.iscas.ac.cn).

available-bw of his surrounding networks without any prerequisite, the traceroute probe for topology discovery and the dispersion technique for available-bw must be combined. The source node not only captures network topology by searching for paths with traceroute probes, but it also estimates the link available-bw using probing packet trains. As a result, in addition to problems faced by existing approaches, we must deal with three extra problems.

The first problem is the effect of reverse path. The accuracy of available-bw measurement heavily relies on the traffic conditions of network paths. Both probing packets that move in forward path and responded ICMP packets that traverse in reverse path may be distorted by network noises. Pathologies such as out-of-order delivery, packet replication and corruption, exist in both directions of end-to-end paths [23], [24]. Thus, based on a single source node, we have to pay more effort so as to reduce the effect of cross traffic in the reverse paths.

The second problem is the dynamics of router delay. In order to precisely monitor the dispersion of packet trains, a probing scheme typically requires timely router responses [17]. However, when a response from a specific router is delayed, it is hard to identify whether the delay is caused by bandwidth or by that router. Furthermore, available-bw measurement tends to be more sensitive to inconstant router delay in the presence of higher bandwidth networks.

The third problem is the path selection. As the scale of topology grows, the number of involved paths increases as well. Since measuring the link available-bw of a path requires more probing packets and more control than finding out the nodes and links of that path, it is necessary to avoid probing redundant paths. This requirement is critical if the source node can only access the Internet through low-bandwidth links.

In this paper, we present RichMap, an original active probing system that can accurately and efficiently measure link available bandwidth on the network topology from an arbitrary end node. Instead of mapping the network topology in an original manner, RichMap adopts heuristics from earlier research on topology discovery with slight modification, and it mainly focuses on measuring the effective range of link available-bw on a given network topology.

The basic idea of available-bw measurement is to search for rate ranges to respectively represent the available-bw of each link. As supported by our measurement methodology, the dispersion of a long packet train will be expanded when it enters a link whose available-bw is lower than its data-rate. Therefore, RichMap sends packet trains at different rates into a path, and monitors per-link dispersion of the trains to narrow the rate ranges for each link available-bw step by step.

To weaken the effect of reverse path and the dynamics of router delay, as well as to optimize the path selection process, many implementation features were added to RichMap. Thereafter, the accuracy and efficiency of RichMap have been verified in both simulations and the network experiments. We analyzed the experiments, and found that RichMap is able to accurately and efficiently capture link available-bw on a topology when most of the nodes are no more than six links away from the source node. Additionally,

though the main purpose is measuring available-bw instead of mapping topology, RichMap can obtain comparatively complete topologies of both small-scale and medium-scale networks.

The rest of this paper is organized as follows. Section 2 discusses the related work. Section 3 presents the measurement methodology, and Section 4 gives some implementation features in details. Section 5 verifies RichMap with simulations and network experiments, and Section 6 analyzes some properties of the resulted maps. Finally, Section 7 concludes the paper.

## II. RELATED WORK

Along two separate research lines, a large number of active probing methods and systems have been introduced to discover the network topology and to capture available-bw, respectively.

Pansiot and Grad first attempted to build a map of the topology by tracing paths to 5000 destinations from a single source node [22]. Siamwalla et al. proposed some heuristics for inferring the topology based on primitives like ping and traceroute [26]. Similarly, Burch and Cheswick discovered and visualized the topologies of some large-scale networks [3]. A step further, Spring et al. attempted to discover the topologies of ten diverse ISPs using over 750 public traceroute servers as measurement vantage points [27]. However, all these methods maintain the destinations addresses as a prerequisite regardless of local network environment, so one cannot expect a satisfying topology from an arbitrary network location.

The first representative system that can build the topology from a source node is Mercator [13]. Mercator largely utilizes the traceroute probes to map the topology. Furthermore, Mercator searches for routers that support IP source-route options from the discovered routers, and then it directs some traceroute probes via these source-route capable routers so as to discover cross-links that otherwise might not be detected. In almost the same way, RichMap generates a list of destination addresses for subsequent probing by analyzing IP prefixes. But RichMap mainly focuses on characterizing the topology and providing detailed information about link available-bw.

From a different perspective, the NetInventory system is designed to discover physical topology in heterogeneous IP networks [1]. A physical topology generally covers more fractions of the networks than an IP-level topology because physical topology captures the complex interconnections of layer-2 network elements, e.g. switches and bridges. Though NetInventory solely relies on SNMP Management Information Base (MIB) [6] that is widely supported in modern IP networks, it is hard for NetInventory to scale because the physical infrastructure that lies under an IP-level topology generally involves many more nodes and communication lines than its upper-layer views.

Besides exploring real network topologies, researchers have designed some virtual models to construct the topology. Waxman introduced what appears to be one of the popular network models [30]. Waxman graphs are generated probabilistically considering the nodes as points in a

Euclidean space. Calvert et al. discussed different graph-based models for representing the topology of large networks, and they focused particularly on aspects of locality and hierarchy [4]. Zegura et al. introduced a comprehensive graph model that included several earlier models, and combined some simpler topologies in a hierarchical structure [31]. Particularly, Faloutsos et al. proposed several empirical power laws that can characterize the Internet inter-domain topology [11]. This further leads to a large amount of research effort in building and analyzing topology models [2], [7], [28].

Besides topology discovery, available-bw measurement is a problem that has attracted the attention of many researchers. The first tool that attempted to measure available-bw was Cprobe [5]. Cprobe transmits short sequences of ICMP echo packets to destination node in a back-to-back fashion, i.e. as close as possible, and calculates the achieved throughput from the timing interval between the first and the last ICMP replies to estimate available-bw. The underlying assumption is that the dispersion of a long packet train is inversely proportional to available-bw. However, in [9], Dovrolis et al. demonstrated that this is not the case. What the dispersion of back-to-back packets captures is the asymptotic dispersion rate (ADR), instead of available-bw.

After Cprobe, on one hand, many techniques that relied on both end-points of a network path were introduced. A typical path available-bw measurement tool, called Pathload, was proposed in [15], and was further explained in [16]. Pathload does not report a single figure; instead, it outputs a rate range in which path available-bw may reside. In order to locate and narrow the rate range, Pathload sends packet streams at different rates and monitors their one-way transmission delay at the destination node. Similarly, RichMap manipulates rate ranges by probing the targeted path with packet trains sent at different rates. But RichMap is deployed in only the source node, and what it captures is link available-bw. Specifically, RichMap captures the per-link dispersion of probing packet trains, instead of one-way transmission delay, to identify the relation between link available-bw and the probing rates.

On the other hand, many methods have also been proposed to estimate bandwidth from the viewpoint of a single source node. However, these methods mainly focused on measuring capacity, i.e. the maximum data rate that the link or path can ever achieve, instead of available-bw. For example, pathchar [10], and the tailgating technique [19] measure link capacity, while Bprobe [5], nettimer [20], pathrate [9], and the PBM methodology [24] measure the end-to-end path capacity.

Recently, Hu et al. addressed the problem of bottleneck location and presented a tool – Pathneck – to infer the location [14]. Pathneck relies on the fact that cross traffic interleaves with probing packets on the links along the path, thus changing the length of the packet train. By measuring the per-link train length, the position of bottleneck link can be inferred. The TTL setting of the packet train in our measurement methodology is similar to the recursive packet train adopted by Pathneck. However, as Hu also noted, Pathneck cannot estimate available-bw because it does not precisely control the inter-packet gap.

Our earlier work in [33] and [34] presented a technique to measure three properties (i.e. location, capacity, and available-bw) of bottleneck link. But what the technique

concerns about is the bottleneck link, instead of all links along the path. Furthermore, it is hard to apply this technique to the whole topology because that requires the precise knowledge of all link capacities. In contrast, RichMap does not refer to available-bw as a single number, but adopts the notation of rate range in which available-bw locates. Through monitoring the dispersion of probing packet trains, RichMap captures link available-bw on the surrounding networks.

### III. MEASUREMENT METHODOLOGY

Our measurement methodology is designed to answer a question: how to measure almost all link available-bw on a given topology? To do this, we first characterize the relation between a probing packet train and link available-bw. Then we design an active probing scheme to capture narrowed rate ranges respectively for each link available-bw. Finally, we illustrate how to estimate the per-link dispersion of a packet train since monitoring the dispersion is an essential part of our scheme.

#### A. Analytical Model

Formally, an end-to-end path is a sequence of First-Come First-Served store-and-forward links that transfer packets from source node  $R_0$  to destination node  $R_n$  through intermediate nodes  $R_1, R_2, \dots, R_{n-1}$ . Link  $L_i = (R_{i-1}, R_i)$  is the data connection from  $R_{i-1}$  to  $R_i$ . Two critical properties of  $L_i$  are link capacity ( $C_i$ ) and link available-bw ( $A_i$ ).  $C_i$  refers to the maximum data rate that  $L_i$  can ever achieve;  $A_i$  is the spare bandwidth that is not utilized by cross traffic, which travels on  $L_i$  at rate  $r_c^i$ , and  $r_c^i = C_i - A_i$ . Compared with the time (a few seconds) our measurement generally takes to measure link available-bw, the path properties can be viewed as constant since they do not change much on the scale of hours [32]. We assume that  $C_i$ ,  $A_i$  and  $r_c^i$  are constant during a measurement process if they are not affected by the probing packet trains sent by  $R_0$ . In addition, path available-bw ( $A$ ) refers to the minimum link available-bw.

Consider that from an arbitrary time instant,  $R_0$  transmits a train of  $N$  packets to  $R_n$  along network path  $P$  at data-rate  $r_p$ . All packets are of size  $S$  and are equally spaced. Fig. 1 shows the train as it traverses on  $L_i$ .

Note that  $\Delta_i$  is the dispersion between the head and tail packets, and per-packet dispersion (PPD) in  $L_i$  is

$$p_i = \frac{\Delta_i}{N(i)-1}, \quad (1 \leq i \leq n). \quad (1)$$

Here,  $N(i)$  is the number of packets that the train maintains when it traverses on  $L_i$ .

Each time after probing the path with a packet train,  $R_0$

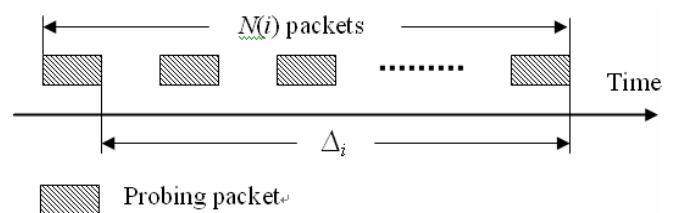


Fig. 1. A probing packet train on  $L_i$ .

collects a sequence of PPD, i.e.  $p_1, p_2, \dots, p_n$ , to identify the relation between  $r_p$  and  $A$ . If  $n = 1$ , the path consists of only one link, so it is not necessary to analyze  $p_1$  because  $p_1$  is solely limited by the spare transmission ability of  $R_0$ , i.e.  $A_1$ .

Now, we consider what happens when  $n > 1$ . Note that the source cannot send packets at a rate that is higher than  $A_1$ , so  $r_p \leq A_1$ . Therefore,  $r_c^1 + r_p \leq r_c^1 + A_1 = C_1$ . As a result,  $L_1$  can carry all the traffic without queueing. This indicates that  $r_c^1$  remains unchanged since the cross traffic on  $L_1$  is not affected by the probing train. In addition, because the cross traffic on  $L_2$  comes from  $L_1$  and elsewhere through  $R_1$ ;  $r_c^2$  remains unchanged because the path properties are assumed to be constant during a measurement process and  $r_c^1$  is not changed.

*Proposition 1:* if  $r_p \leq A$ , then  $\forall i \in (1, n], p_i = p_{i-1}$ .

First, let's analyze how the train enters  $L_2$  from  $L_1$ . In a  $p_1$  period, a probing packet enters  $L_2$ . During the same time, the amount of cross traffic that enters  $L_2$  via  $R_1$  is  $X_2 = r_c^2 \cdot p_1$ . Thus, the total amount of packets that  $L_2$  accepts during a  $p_1$  period is  $S + X_2$ . Since  $r_p \leq A$  and  $A$  equals to the minimum link available-bw, we have  $r_p \leq A \leq A_2$  and

$$S + X_2 = (r_p + r_c^2) \cdot p_1 \leq (A_2 + r_c^2) \cdot p_1 = C_2 \cdot p_1 \quad (2)$$

As a result,  $L_2$  can carry forward all the packets from  $R_1$  without queueing. Therefore, the packet train is still transmitted at rate  $r_p$  on  $L_2$ , and  $p_2 = p_1$ . Obviously, the cross traffic on both  $L_2$  and  $L_3$  is not affected by the packet train.

Then, we inductively prove this in the subsequent links. Suppose  $n > 2$ ,  $p_i = p_{i-1}$  for  $i = 2, 3, \dots, k$  ( $2 \leq k \leq n-1$ ). The packet train on  $L_k$  travels at rate  $r_p$ ; both  $r_c^k$  and  $r_c^{k+1}$  are not altered.

As the train enters  $L_{k+1}$  from  $L_k$ , only a probing packet moves into  $L_{k+1}$  in a  $p_k$  period, and the amount of cross traffic that goes into  $L_{k+1}$  from  $R_k$  in that  $p_k$  period is  $X_{k+1} = r_c^{k+1} \cdot p_k$ . Consequently, the total amount of packets that  $L_{k+1}$  accepts during a  $p_k$  period is  $S + X_{k+1}$ . Because  $r_p \leq A \leq A_{k+1}$ ,

$$S + X_{k+1} = (r_p + r_c^{k+1}) \cdot p_k \leq (A_{k+1} + r_c^{k+1}) \cdot p_k = C_{k+1} \cdot p_k \quad (3)$$

Thus, all the incoming packets can be transmitted by  $L_{k+1}$  without being queued, so  $p_{k+1} = p_k$  and the rate of the train on  $L_{k+1}$  is still  $r_p$ ; additionally,  $r_c^{k+1}$  and  $r_c^{k+2}$  remain unchanged.

Finally, we conclude that  $\forall i \in (1, n], p_i = p_{i-1}$ .

*Proposition 2:* if  $r_p > A$ ,  $\exists i \in (1, n], p_i > p_{i-1}$ .

Since  $r_p > A$ , there must be at least one link available-bw that is lower than  $r_p$ . Specifically, we name a link *the first narrow link* if its available-bw is first less than  $r_p$ . Obviously,  $L_1$  cannot be the first narrow link because we always have  $r_p \leq A_1$ .

Now suppose that  $L_2$  is the first narrow link, then  $r_p > A_2$ . As we have shown above, when the train traverses from  $L_1$  to  $L_2$ , the total amount of traffic that enters  $L_2$  in  $p_1$  is  $S + X_2$ , so

$$S + X_{k+1} = (r_p + r_c^{k+1}) \cdot p_k \leq (A_{k+1} + r_c^{k+1}) \cdot p_k = C_{k+1} \cdot p_k \quad (4)$$

This indicates that  $L_2$  has to take more time to carry out the total traffic it accepts in a  $p_1$  period. Therefore, queue is built up and dispersion of the packet train is enlarged. Specifically,  $L_2$  expands the PPD to be

$$p_2 = \frac{S + X_2}{C_2} \quad (5)$$

According to (4),  $p_2 > p_1$ .

Now we consider the case that  $L_2$  is not the first narrow link and  $n > 2$ . Assuming that the first narrow link is  $L_k$ ,  $r_p > A_k$  and  $2 < k \leq n$ . Naturally, links  $\{L_1, L_2, \dots, L_{k-1}\}$  can be viewed as a short path  $P_S$  included by the original path  $P$ . In  $P_S$ , all the link available-bw is not lower than  $r_p$ . According to proposition 1, the data-rate of the train on  $L_{k-1}$  is still  $r_p$ , and  $r_c^k$  remains unchanged. The total amount of traffic that travels into  $L_k$  in  $p_{k-1}$  is  $S + X_k$ , and

$$S + X_k = (r_p + r_c^k) \cdot p_{k-1} > (A_k + r_c^k) \cdot p_{k-1} = C_k \cdot p_{k-1} \quad (6)$$

$L_k$  needs more than  $p_{k+1}$  to carry out the traffic, and

$$p_k = \frac{S + X_k}{C_k} \quad (7)$$

According to (6),  $p_k > p_{k-1}$ .

Finally, we conclude that if  $r_p > A$ ,  $\exists i \in (1, n], p_i > p_{i-1}$ .

## B. Active Probing Scheme

Based on the analytical model, we come to capture the rate ranges for each link available-bw through monitoring the PPD sequence of packet trains sent at different data-rates.

In RichMap, the link available-bw  $A_i$  is not a number, but a rate range  $[R_{\min}, R_{\max}]$  in which  $A_i$  resides. The upper bound  $R_{\max}$  is the lowest  $r_p$  that is identified to be higher than  $A_i$ , and the lower bound  $R_{\min}$  is the highest  $r_p$  that is proved to be lower than or equal to  $A_i$ . Initially,  $R_{\min} = 0$  and  $R_{\max}$  is marked as unknown. At a probing rate  $r_p$ , the source node probes the path with a long packet train, and then it collects the PPD sequence. After each probe, all rate ranges are adjusted (Fig. 2).

Specifically,  $L_i$  expands PPD if  $p_i > p_{i-1}$  is detected. Note that when  $p_i = p_{i-1}$  and  $L_i$  is not the *first narrow link*, we cannot set  $R_{\min}$  of  $L_i$  to  $r_p$ . The reason is that under such a situation, the probing packet train must enter  $L_i$  at an unknown data-rate that is lower than  $r_p$ . There is not a rate range for  $A_i$  because  $L_i$  is directly attached to  $R_0$  and  $A_i$  can be directly estimated by  $R_0$ .

In addition to obtain the rate range for link available-bw, another critical problem is how to schedule  $r_p$ . Initially,  $r_p = A_1$  because  $A_1$  determines the maximum  $r_p$  that  $R_0$  can obtain. After a probe, rate ranges are updated, and the next packet train will be sent at  $r_p = (r_p - \delta)$ . If  $r_p \leq \delta$ , or  $r_p$  is lower than the lower bounds of all rate ranges, or no PPD expansion occurs, the process terminates; here  $\delta$  is the bandwidth resolution that decides the precision of probing rates. This process generally stops after probing the path about  $A_1 / \delta$  times.

```

01: while ( $r_p > \delta$ )
02: {
03:   probe_and_collect_ppd(); // packet train sent at  $r_p$ 
04:   ppd_expanded = false; // no PPD expansion yet
05:   for ( $i = 2; i \leq n; i++$ ) // for links  $L_2, L_3, \dots, L_n$ 
06:   {
07:     if ( $p[i] > p[i-1]$ ) //  $L_i$  expands the PPD?
08:     {
09:       if ( $R_{\max}[i] > r_p \parallel R_{\max}[i] = \text{UnKnown}$ )
10:          $R_{\max}[i] = r_p$ ; // reset the upper bound of  $L_i$ 
11:       ppd_expanded = true;
12:     }
13:     else if ( $p[i] == p[i-1]$ ) // no PPD expansion?
14:     {
15:       if ( $R_{\min}[i] < r_p \ \&\& \ !\text{ppd\_expanded}$ )
16:          $R_{\min}[i] = r_p$ ; // reset the lower bound of  $L_i$ 
17:     }
18:   }
19:   if ( $r_p < \text{Min}_{2 \leq i \leq n}(R_{\min}[i])$  or  $\! \text{ppd\_expanded}$ )
20:     exit(); // exit this measurement process
21:   else
22:      $r_p = r_p - \delta$ ; // reset  $r_p$ 
23: }

```

Fig. 2. Pseudo code for adjusting per-link rate range.

Finding a suitable scheduling algorithm for  $r_p$  that can work under most network situations is a very intriguing task because it is the outcome of many tradeoffs. In our experiments, the algorithm (Fig. 2) exhibited two advantages. First, it enables the probing process to converge after about  $A_1 / \delta$  probes without significant loss in accuracy. Second, the precision of the rate ranges can be adjusted simply by adjusting bandwidth resolution  $\delta$ . This is very important for various users since they typically require diverse fidelities and performances.

### C. Capturing Per-link PPD

To identify the relation between available-bw and  $r_p$ ,  $R_0$  must be able to obtain the PPD sequence. Based on the ICMP mechanism [25] supported by the Internet infrastructure, we build up a probing packet train that can invoke desired router responses. Specifically,  $2N$  IP packets of size  $S$  are sent to the destination at data-rate  $r_p$ . These packets are ordered in two groups ( $G_1$  and  $G_2$ ),  $G_1$  and  $G_2$  are separated by  $\Delta G$ . Packets in each group are equally spaced by  $\Delta P$ . Obviously,  $\Delta P = p_1$ , and

$$r_p = \frac{S}{\Delta P} \quad (8)$$

The TTL fields of the  $N$  packets in  $G_1$  are set to  $\{1, 2, \dots, (n-1), n, n, \dots, n\}$ ; and TTL of the  $N$  packets in  $G_2$  are set to  $\{n, n, \dots, n, (n-1), \dots, 2, 1\}$ . Here,  $n$  is the number of links on a targeted path, and  $N > n$ . Fig. 3 illustrates a packet train built to probe a 7-link path.

Setting TTL fields in such a way makes each router along the path reply two ICMP packets back to  $R_0$  as the train traverses through. When the train arrives at the first router  $R_1$ , its head and tail packets expire since their TTL values are 1 (TTL expires). As a result, the two probing packets are dropped and  $R_1$  sends two ICMP packets back to  $R_0$ . The other packets of the train are forwarded to  $R_2$  after their TTL

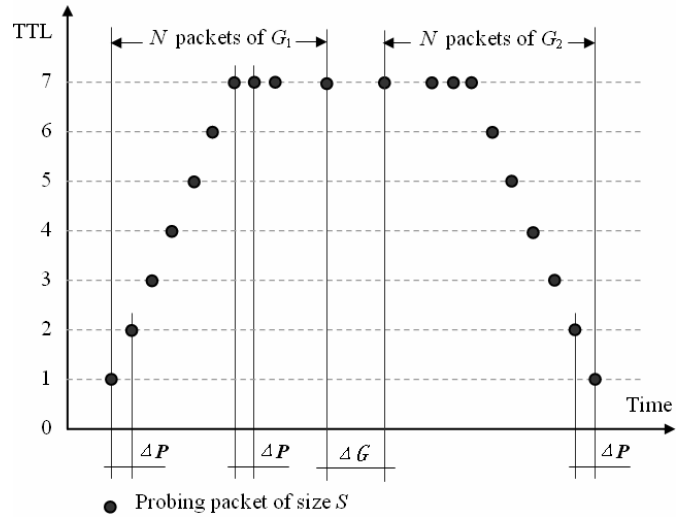


Fig. 3. A packet train for a 7-link path.

are respectively decreased by 1. In the same way, each subsequent router repeats the above process.

Therefore, every intermediate router returns two ICMP packets to the source. The source then measures dispersion between the arrivals of two ICMP packets from a router to estimate the dispersion of packet train in the incoming link of that router. Equation (1) is then rewritten as follows:

$$p_i = \frac{\Delta_i - \Delta G + \Delta P}{2N - 2 \cdot (i-1) - 1}, \quad (1 \leq i \leq n). \quad (9)$$

Note that the ICMP packets returned by  $R_1, R_2 \dots R_{n-1}$  are ICMP time-exceeded error packets, while what  $R_n$  returns are ICMP destination-unreachable error packets because the port numbers of the IP packets are set to an abnormally high integer.

## IV. IMPLEMENTATION FEATURES

The implementation of RichMap executes two threads. One generates a list of destination addresses, probes these addresses for path information, and inserts the path into the topology if the path is not detected before. The other obtains a latest copy of the topology, selects paths out from the copy, measures all link available-bw of these paths, and refreshes the topology with up-to-date link available-bw. RichMap is a collection of about 15,000 lines of C++ code and 1000 lines of Perl code.

In real networks, there are too many factors that might affect the measurement process. To ensure the accuracy and efficiency of RichMap, in addition to some common heuristics introduced by other methods, e.g. controlling the probing rate  $r_p$  [15], resolving IP address alias [13], and identifying nodes in the same network [27], several important features are added to RichMap.

Our measurement methodology is designed to answer a question: how to measure almost all link available-bw on a given topology? To do this, we first characterize the relation between a probing packet train and link available-bw. Then we design an active probing scheme to capture narrowed rate ranges respectively for each link available-bw. Finally, we illustrate how to estimate the per-link dispersion of a packet train since monitoring the dispersion is an essential part of our scheme.

A. Probing Informed Random Addresses

A classic problem that single-source topology discovery systems must deal with is how to obtain reachable destination addresses in the absence of external information. Instead of using addresses randomly chosen from the entire IP address space [26], to adapt to the local network environment quickly, we choose the informed random address algorithm proposed in [13] with slight modification.

The basic idea of this algorithm is to guess the IP address prefixes that contain reachable nodes with two basic techniques. First, whenever  $R_0$  receives an ICMP response from address  $A$ , some prefixes of  $A$  may contain addressable nodes. Second, if  $R_0$  has an addressable prefix  $P$ , the neighboring prefixes of  $P$  may also contain addressable nodes.

However, our testing engineers frequently complained that the spanning process for prefixes is very time-consuming when  $R_0$  is located inside a Local Area Network (LAN). The reason is that the above algorithm chooses the IP address of  $R_0$  as the seed to start prefix searching, but in the LAN this IP address is generally a private address [12]. Undoubtedly, in such a case, iterating the algorithm to find prefixes for a public IP address is tedious. Furthermore, this problem is urgent since public IP addresses have become a kind of scarce resources and many applications are used to be deployed within LANs.

To solve this problem, we slightly modify the method for seed selection. Basically, if  $R_0$  is probing the Internet from an interface that is assigned a public IP address, RichMap explores the prefixes in the default manner. When the outgoing interface of  $R_0$  is assigned a private address, RichMap either adopts the IP address of  $R_0$ 's Domain Name Service (DNS) server as the seed, or it probes preconfigured IP addresses (e.g. the IP of *www.ieee.org*), and starts prefix exploration from the source IP addresses of returned ICMP packets. In our experiments, this enhancement enables RichMap to scan the surrounding networks efficiently.

Finally, to construct the topology,  $R_0$  repeatedly selects a prefix, uniformly selects an address  $A$  from within that prefix, and probes the path to address  $A$  with traceroute primitives. If  $A$  is reachable, such a probe generally results in a sequence of routers  $R_1, R_2, \dots, R_n$ . RichMap then inserts nodes  $R_1, R_2$ , etc. and links  $R_0-R_1, R_1-R_2, R_2-R_3$ , etc. into its map if these nodes and links are not in the map.

B. Path Selection and Preparation

As a thread discovers the topology, another thread obtains a copy of the latest topology, selects targeted paths from the copy, and measures the link available-bw of these paths. Selecting paths from the topology for available-bw measurement is an important problem. Because there are generally too many paths inside a topology and short paths are included by long paths, it is difficult to cover all link available-bw by measuring only a few paths.

RichMap applies a hop-limited path selection algorithm. The basic idea is to select a path that starts from  $R_0$ , crosses exactly  $H$  links, and ends at a node in the topology. Initially,  $H$  is set to 1. Each time after probing all the  $H$ -link paths that starts from  $R_0$ , the value of  $H$  is increased by 1. If  $H$  exceeds the maximum hop number of the topology or all links on the topology are covered, the measurement process stops and the

thread requests for the latest copy of topology to start another measurement process again. In this way,  $R_0$  captures the available-bw of nearest links first. Additionally, the nearer a link to  $R_0$ , the more frequently its available-bw would be refreshed. In most of the cases, this is what the users expect because they are more interested in their neighboring elements than the remote elements.

Before capturing the link available-bw of a path, it is necessary to capture several path properties, including  $A_1$  that decides the maximum  $r_p$  that  $R_0$  can achieve, and round-trip time (RTT) between  $R_0$  and  $R_n$ . This is done by sending out a set of packets to  $R_n$  as soon as possible, recording the time that a probing packet is sent and an ICMP packet is received. Path properties should be obtained at initial phase because they help to discover route change and to construct the probing packet trains in later phrase.

C. Capturing  $P_n$  and  $P_0$

When measuring the available-bw, we simply cannot expect that every probing packet sent to  $R_n$  can invoke a reply because  $R_n$  may generate a limited number of ICMP packets in a constant interval. In fact,  $R_n$  will typically only generate ICMP packets for some of the probing packets. As a result, the source has to carefully manipulate the timing of ICMP packets returned from  $R_n$  to estimate  $p_n$ .

The source records the moment that it receives the first ICMP packets from  $R_n$  for the packet groups  $G_1$  and  $G_2$  respectively:  $\{t_1, t_2\}$ . For example, the  $m_1^{th}$  packet of  $G_1$  and the  $m_2^{th}$  packet of  $G_2$  respectively trigger the first ICMP packets from  $R_n$  for the two groups, so  $\Delta_n = t_2 - t_1$  and

$$p_n = \frac{\Delta_n - \Delta G + \Delta P}{N - m_1 + m_2} \tag{10}$$

Note that if a whole group of packets do not trigger any ICMP packet from  $R_n$ , we try again after enlarging  $\Delta G$  because sometimes  $R_n$  cannot generate more than one ICMP error packet in a very short interval.

An additional PPD ( $p_0$ ) is added to the PPD sequence to estimate the packet dispersion at the exit of  $R_0$ . In the analytical model, we have  $r_p \leq A_1$ . However, in real networks, this is not always the case.  $R_1$  sometimes rejects packets sent by the  $R_0$  due to the overflow of incoming queue. This indicates that  $A_1$  not only depends on how fast  $R_0$  can transmit a train, but also relies on how fast  $R_1$  can accept the train. This is a common feature of all links. The processing ability of  $R_1$  indeed limits  $A_1$  and  $r_p$ . By taking in  $p_0$  as the PPD in link " $L_0$ " that virtually locates inside  $R_0$ , we enhance the ability of detecting PPD expansion. Specifically,

$$p_0 = \Delta P \tag{11}$$

D. A Fleet of Packet Trains

Recall that when a packet train enters a congested link where the link available-bw is not high enough to embrace the train without queueing, the PPD expansion takes place. At each probe, sometimes it is inaccurate to identify that a link (e.g.  $L_i$ ) expands the PPD using the equation " $p_i > p_{i-1}$ " because the PPD sequence may be skewed by network noises. Therefore, to accurately identify the PPD expansion, a fleet of  $F$  packet trains are applied for each  $r_p$ .

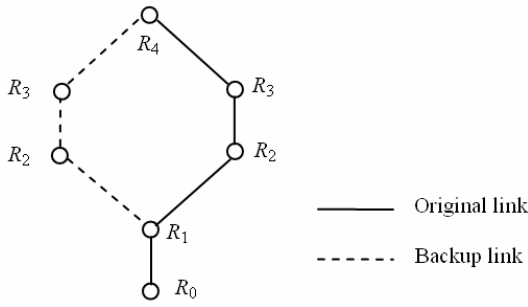


Fig. 4. Probing packets go through two paths from  $R_0$  to  $R_4$ .

Generally, there are two kinds of expansions: sharp and gradual expansions. The sharp expansion denotes the situation that PPD is dramatically enlarged by a specific link. While gradual expansion refers to the situation that there is no significant difference between neighboring PPDs, but PPDs in downstream links are obviously larger than that in upstream links. Every packet train results in a sequence of  $n+1$  PPD for  $L_0, L_1, \dots, L_n$ , a fleet of  $F$  packet trains leads to PPDs  $\{p_i^1, p_i^2, \dots, p_i^F\}$  for  $L_i$ . Let  $M(i)$  denotes the mean value of the  $F$  PPDs for  $L_i$ . Taking only the mean PPD in each link enable us to remove the outliers and focus on the regular values. Now,  $L_i$  causes a sharp expansion if

$$M(i) - M(i-1) \cdot \lambda > 0 \quad (12)$$

Frequently, although there is not a sharp expansion detected, the path can observe a gradual expansion by comparing all the PPDs as follows:

$$GE = \frac{\sum_{i=1}^n I[M(i) - M(i-1) \cdot \mu]}{n} \quad (13)$$

Here  $I(x) = 1$  if  $x > 0$ , or else  $I(x) = 0$ . The hard gauge parameter  $\lambda$  is larger than the soft gauge parameter  $\mu$ ; if  $GE \geq 0.33$ , a gradual expansion takes place. If a sharp expansion is detected,  $R_0$  adjusts the rate ranges as shown in Fig. 2. When the gradual expansion is identified,  $r_p$  is decreased by  $\delta$ , leaving all the rate ranges unchanged.

Additionally, there is another metric that can be used to identify the gradual expansion: the loss proportion. Since a router often drops packets when it has not enough space to accept the probing packet trains, the loss of probing packets can also be considered as an evidence for dispersion expansion. Specifically, a probing packet is defined as loss if the source does not receive a corresponding ICMP packet. For a single train, the loss proportion ( $L_p$ ) is the number of loss probing packets ( $N_L$ ) over the number of probing packets ( $2N$ ):

$$L_p = \frac{N_L}{2N} \quad (14)$$

If  $L_p \geq \beta$  and no sharp PPD expansion detected,  $R_0$  adjusts  $r_p$  as if the gradual expansion occurs. Generally,  $0 < \beta < 1$ .

### E. Other Considerations

During the measurement,  $A_1$  may limit what the rate ranges can cover. For example, if  $A_i > A_1$ , the upper bound of the rate range for  $L_i$  cannot be inferred since  $L_i$  always accepts the packet train without causing PPD expansion. Furthermore, because the bottleneck link limits the transmission rates of packet trains that traverse through, RichMap sometimes

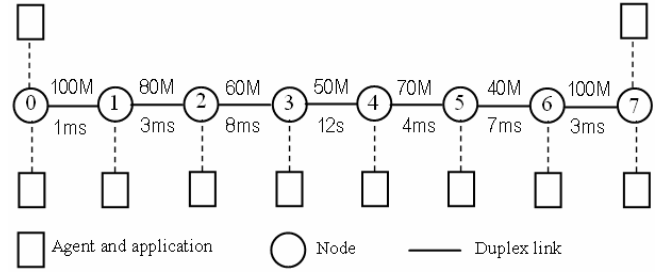


Fig. 5. Simulation topology.

cannot clearly infer the available-bw of links located right after the bottleneck link.

The topology cannot be static since route changes may occur at any time. When the available-bw measurement thread obtains the latest copy of the topology and start probing, it may find that some paths have been changed. The thread detects the change by checking whether the returned ICMP packets come from desired routers or not. If the measurement thread finds new links (also called backup links), it directly adds these links to the topology (Fig. 4).

Though discovering a complete topology is not our purpose, measuring available-bw on a complete topology can give users a sound view of their surrounding networks. Because RichMap attempts to discover the topology from a single network node, it may miss some links that are unreachable from  $R_0$ . To find these links, RichMap first searches for some routers that support source-route option, and then it probes the already discovered routers via these source-route capable routers. In [13], Govindan et al. exhibits more details about this technique and proves that 90% links can be found if only 5% routers are source-route capable. In our network experiments, we found this technique did work well.

## V. VERIFICATION

The accuracy and efficiency of RichMap have been verified through simulations and network experimtns. First, a simulation environment is built to study how RichMap measure all link available-bw on a path. Second, RichMap is compared with Pathload on the ability of measuring path available-bw. Finally, besides available-bw measurements, the topologies that RichMap constructed for 12 diverse networks are also evaluated.

### A. Simulation Verification

The ability of RichMap on measuring link available-bw on a path has been verified in Network Simulator (NS) [21], which is a controlled and reproducible network simulating environment. Specifically, RichMap was implemented in both the application and agent levels of NS. A linear topology is used throughout the simulation (Fig. 5). Nodes 0 and 7 are the source node ( $R_0$ ) and the destination node ( $R_7$ ), respectively; nodes 1-6 are intermediate nodes. All links are duplex; capacities of links are in the unit of bits per second. In addition, every link applies the drop-tail queuing principle. Table I lists the parameters used by RichMap.

Before our verification, we slightly modify the ICMP mechanism of NS. In real networks, it is the node (router) that counts the TTL field of each IP packet, and responses with ICMP error packet if the TTL expires. While in NS, it is the

TABLE I  
PARAMETERS

Parameter	Description	Default
$N$	The number of packets in half a train.	30
$F$	The number of packet trains in a fleet.	8
$S$	The size of the probing packet.	300 Bytes
$\delta$	The bandwidth resolution.	$A_1/12$
$\lambda$	The hard guage parameter (Section IV).	0.15
$\mu$	The soft guage parameter (Section IV).	0.05
$\beta$	The loss guage parameter Section IV).	0.25
$u_i$	The utilization of a path; for link $L_i$ ( $1 \leq i \leq n$ ), $r_c^i / C_i = u_i$ .	Set according to scenarios

link that decreases TTL and simply drops the expired packets without generating any ICMP packet. To walk around this, we developed an ICMP application and attached it to every node except  $R_0$ . The ICMP application responds with a 56-byte ICMP packet back to  $R_0$  whenever it receives a probing packet. Thus,  $R_0$  directly sends packets to the ICMP application attached to a node, if  $R_0$  wants that node to respond with ICMP packets.

In the following simulations, we apply one hop persistent (OHP) cross traffic to the path. Specifically, an OHP packet stream comes from four CBR traffic sources, and must exit the path after traversing only one link. Packets of each OHP traffic stream are carefully set as follows: 15% 576 bytes, 20% 1500 bytes, 50% 40 bytes and 15% randomly distributed between 40 and 1500 bytes, similar to the Internet packets measured in [29]. All links are equally utilized by OHP cross traffic. For example, the path is 40% utilized means that all links in the path are 40% utilized, i.e. the cross traffic on each link is {40, 32, 24, 20, 28, 16, 40Mbps}. Actually, there are two paths in Fig. 5: the forward path from node 0 to node 7, and the reverse path from node 7 back to node 0. What we are measuring is the link available-bw on the forward path.

Table II shows how the rate ranges for all link available-bw are adjusted in a single measurement process. In this case,  $u_i = 40\%$ . The actual link available-bw  $\{A_1, A_2 \dots A_7\}$  are {60, 48, 36, 30, 42, 24, 60Mbps}. At the first probe,  $r_p = 60\text{Mbps}$ , both  $L_2$  and  $L_3$  cause sharp PPD expansions; after setting the upper bounds of  $A_2$  and  $A_3$  to 60Mbps,  $R_0$  decreases  $r_p$  by  $\delta = 5\text{Mbps}$  and probes again. Finally, at the 9<sup>th</sup> probe,  $r_p = 20\text{Mbps}$ , the lower bound of  $A_7$  is identified to be 20Mbps because it is the first time that both  $L_7$  and the upstream links do not expand the PPD. Because the lower bounds of all rate ranges are higher than  $r_p$ , the measurement process terminates. Note that there is not a rate range for  $A_1$  because, in the preparation phrase,  $A_1$  is directly worked out by testing how fast  $R_0$  can transmit packets in a limited interval without affecting other data flows on  $L_1$ .

A step further, we evaluate RichMap under different load

conditions. The reverse path is 40% utilized, while the forward path is 20%, 40%, and 60% utilized, respectively. Fig. 6 shows the measured rate ranges for all links. Particularly, “FP  $u_i = x\%$ ” means the forward path is  $x\%$  utilized.

The first observation is that RichMap is able to accurately capture the link available-bw, no matter if the targeted path is congested ( $u_i = 60\%$ ) or underutilized ( $u_i = 20\%$ ). The rate ranges for  $A_2, A_3, A_4$ , and  $A_6$  focus on the actual available-bw. Moreover, RichMap can always precisely identify the rate range for bottleneck link ( $L_6$ ). Though the rate ranges for  $A_5$  and  $A_7$  are obscure, they indeed give us a signal that  $A_5$  and  $A_7$  must be higher than the displayed lower bounds.

The second observation is that the rate ranges for  $A_5$  and  $A_7$  provide only lower bounds.  $A_5$  is higher than  $A_3$  and  $A_4$ , and  $A_7$  is higher than  $A_3, A_4$ , and  $A_5$ . Therefore, the upper bounds of rate ranges for  $A_5$  and  $A_7$  left unknown because packet trains cannot transmit on  $L_5$  and  $L_7$  at a rate that is higher than  $A_5$  and  $A_7$ . In addition, packet trains transmitted at a rate  $r_p$  that is higher than  $A_3$  or  $A_4$  must be expanded before they reach  $L_5$  and  $L_7$ . In this case, the packet train reaches  $L_5$  and  $L_7$  at an unknown data-rate that is lower than  $r_p$ , so the lower bounds of  $A_5$  and  $A_7$  cannot be set to  $r_p$ . As a result, these lower bounds can only be set to a value that is not higher than both  $A_3$  and  $A_4$ .

The third observation is that as the path load grows heavier, the resulted rate ranges grow wider as well.  $R_0$  generally cannot identify the sharp PPD expansion when  $r_p$  is just a bit higher than the actual available-bw. Recall that the PPD is expanded when the packet train enters a congested link. If  $L_{i+1}$  is the first link that expands PPD, stated in Section 3, in a  $p_i$  period, totally  $S + C_{i+1}u_i p_i$  amount of

$$p_{i+1} = \frac{S + C_{i+1} \cdot u_i \cdot p_i}{C_{i+1}} \tag{15}$$

Naturally, we have

$$\frac{p_{i+1}}{p_i} = \frac{S/p_i + C_{i+1} \cdot u_i}{C_{i+1}} = \frac{r_p + C_{i+1} \cdot u_i}{C_{i+1}} \tag{16}$$

As revealed by (16), when  $u_i$  is low (20%), a small change in  $r_p$  can leads to large PPD difference. But when  $u_i$  is high enough, e.g.  $u_i = 60\%$ , this is not the case. Thus, as  $u_i$  grows, the upper bounds of rate ranges become hard to identify.

Measurement time is the main indicator of the efficiency of RichMap. Table II lists the time that RichMap takes to measure all link available-bw on the path. Under different load conditions, a single measurement process averagely consumes 2.02–2.83 seconds. In addition, the minimum time that it takes is 1.90–2.52 seconds; and the maximum time that it takes is 2.35–3.11 seconds. The measurement time increases slowly as the path utilization grows from 20% to 60%. For a user or an application, RichMap is efficient enough in measuring all link available-bw on a path.

TABLE II  
THE RATE RANGES FOR ALL LINK AVAILABLE-BW (THE SYMBOL  $U$  REFERS TO "UNKNOWN BANDWIDTH")

Probe #	$r_p$ (Mbps)	Rate Range for $A_2$ (Mbps)	Rate Range for $A_3$ (Mbps)	Rate Range for $A_4$ (Mbps)	Rate Range for $A_5$ (Mbps)	Rate Range for $A_6$ (Mbps)	Rate Range for $A_7$ (Mbps)
1	60	0, 60	0, 60	0, $U$	0, $U$	0, $U$	0, $U$
2	55	0, 55	0, 55	0, $U$	0, $U$	0, $U$	0, $U$
3	50	0, 50	0, 50	0, $U$	0, $U$	0, $U$	0, $U$
4	45	45, 50	0, 45	0, $U$	0, $U$	0, $U$	0, $U$
5	40	45, 50	0, 40	0, $U$	0, $U$	0, $U$	0, $U$
6	35	45, 50	35, 40	0, 35	0, $U$	0, $U$	0, $U$
7	30	45, 50	35, 40	0, 30	30, $U$	0, 30	0, $U$
8	25	45, 50	35, 40	25, 30	30, $U$	0, 25	0, $U$
9	20	45, 50	35, 40	25, 30	30, $U$	20, 25	20, $U$

TABLE III  
RICHMAP MEASUREMENT TIME

FP $u_t$	Measurement Time (Second)		
	Min	Mean	Max
20%	1.90	2.02	2.35
40%	2.06	2.40	2.88
60%	2.52	2.83	3.11

### B. Comparative Evaluation

In addition to simulation, the accuracy of link available-bw is compared with Pathload, which is typically an end-to-end measurement tool that requires the cooperation of  $R_0$  and  $R_n$ . Note that what Pathload measures is the path available-bw, instead of link available-bw. However, because the path available-bw equals to the available-bw of bottleneck link, we compare the minimum link available-bw of RichMap with the results from Pathload so as to indirectly verify the ability of RichMap. We do not attempt to make quantitative conclusions because that demands the deployment of Pathload on a large number of nodes. Instead, our objective is to evaluate RichMap and Pathload in public networks.

To deploy Pathload in some end nodes, we elaborately search in the discovered topologies for nodes on which we have administrative control. We then deployed the receiver part of Pathload on these nodes and run the sender part on  $R_0$  to measure the paths from  $R_0$  to these nodes. The frequency that Pathload and RichMap access the a path is quite different. In our experiment, every instance of Pathload is configured to measure specific paths once per hour, while RichMap averagely access these paths once every 12 hours. We performed the comparison in 46 paths, Fig. 7 typically illustrates the result on a 3-link path and a 6-link path.

We found that RichMap is able to capture the available-bw of bottleneck link as accurately as Pathload does. When a path consists of no more than six links, RichMap outputs rate ranges that exactly focus on the path available-bw. When the number of links increases, the rate ranges tend to enlarge as well. But the rate ranges for the 6-link path is not larger than that for the 3-link path. This is a typical case in our experiments. Because the bottleneck link available-bw of longer paths is generally lower, the rate ranges of bottleneck links are focused even though they often fluctuate due to network noises.

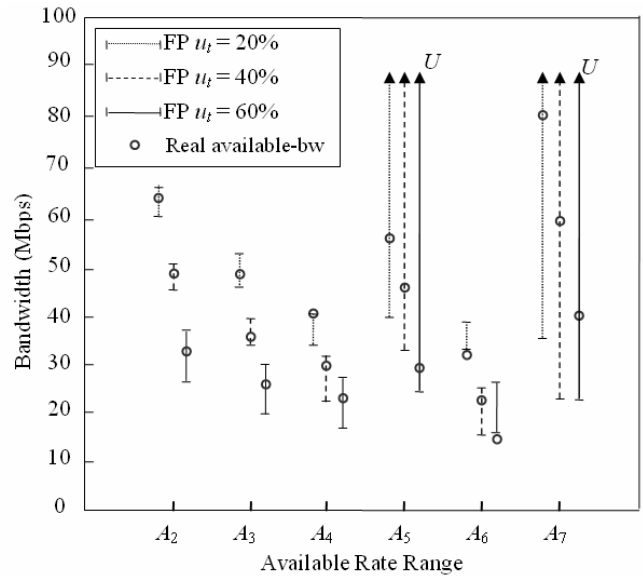


Fig. 6. Link available-bw under different load conditions. The symbol  $U$  refers to "unknown bandwidth".

In addition, RichMap took slightly more time than Pathload. Though the implementation of grey region generally requires more probes before convergence, Pathload was supported by software deployed at both end-points of paths. In contrast, RichMap relied on only the source node, so it sometimes had to wait a while for the ICMP packets returned by designated routers.

### C. Topology Verification

Though discovering a complete topology is not our main purpose, we want to know whether the topologies that RichMap builds for available-bw measurements are complete. From July 2004 until now, RichMap has been applied to measure 12 diverse networks: six small-scale networks that are respectively located in six different buildings, four campus networks, and two large-scale commercial networks. Table IV lists how many nodes and links RichMap have discovered in the 12 networks.

We are encouraged after consulting network operators of the 12 networks<sup>1</sup>. Operators of the six small-scale networks praised our work because our maps captured more than 95% nodes and links, and the maps in fact gave them more

<sup>1</sup>ISCAS, GUCAS, USTC, TSINGHUA, and PKU are the Institute of Software, Chinese Academy of Sciences, the Graduate University of the Chinese Academy of Sciences, University of Science and Technology of China, Tsinghua University, and Peking University, respectively. Beijing Broadband and Beijing Telecom are two commercial ISPs of Beijing, China.

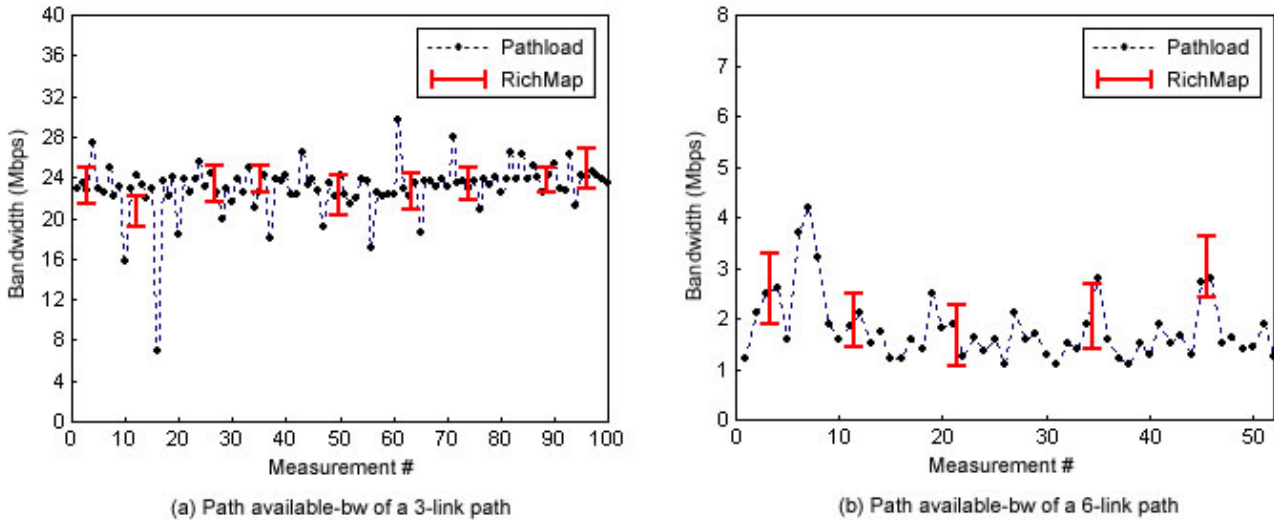


Fig. 7. Path available-bw measurement of RichMap and Pathload (we plot only the mean value of the rate range of Pathload).

insight than network structure. For example, four of the six networks used RichMap to figure out the bottleneck links and upgraded their networks. One operator even redesigned his network structure to optimize the network performance by monitoring the utilization of 28 critical links using RichMap.

Three of the four campuses responded that our topologies captured more than 80% routers. Since each link is marked with its available-bw, the topologies indeed gave a clear insight of their network structure, e.g. the backbones and department networks. The two ISPs answered that more than 50% their nodes have been detected by RichMap. However, both campuses and ISPs did not give any comment on the link connections because either they had not maintained such knowledge or they reserved the link information for security reasons.

In particular, two RichMap instances were separately applied to measure the GUCAS network from two locations: one connected to the backbones and the other stayed inside a department LAN. The two instances resulted in two topologies that shared 94% common nodes and 86% common links. This reveals that, in both small-scale and medium-scale networks (like campus), the topologies obtained by RichMap are stable and complete regardless of the location of source node.

VI. RESULT AND ANALYSIS

After verification, we come to view the maps that RichMap outputs, and to analyze the measurement results. Due to space limitation, only the measurement over GUCAS network is presented. The following analysis can be applied to most of our measurements because the GUCAS network typically consists of high-performance backbones, which connect a large number of diverse and small-scale LANs.

Fig. 8 shows the measured links available-bw over two copies of the topology. Our first observation is that RichMap is able to capture both backbones and small-scale networks that are of different transmission capabilities. There are about ten high-speed links, connecting many local networks. Some local networks are built with high-performance equipments, while many others are not. Here, the source node happens to be in the same LAN of a backbone router. This enables RichMap to probe for link available-bw in a large range of

TABLE IV  
TOPOLOGY INFORMATION (ORDERED BY MEASUREMENT DATE)

#	Network	Scale	No. of nodes	No. of links
1	GUCAS	Campus	251	633
2	USTC	Campus	337	1040
3	GUCAS Building 1	Small-Scale	79	186
4	GUCAS Building 2	Small-Scale	65	190
5	USTC Building 1	Small-Scale	91	350
6	Beijing Broadband	Large-scale	1409	5956
7	TSINGHUA	Campus	482	1614
8	Beijing Telecom	Large-scale	1826	4533
9	ISCAS Building 1	Small-Scale	98	225
10	ISCAS Building 2	Small-Scale	87	201
11	PKU	Campus	322	1530
12	USTC Building 2	Small-Scale	80	193

data-rates because the backbone gives the source node a very large  $A_1$ .

Another observation is that although the available-bw of the backbone links are steady, most link available-bw changes with time. Link available-bw of the 54<sup>th</sup>-hour topology is generally higher than that of the 80<sup>th</sup>-hour topology.

We analyzed the log of RichMap and found that measurement of the 54<sup>th</sup>-hour topology was largely conducted at night, while measurement of the 80<sup>th</sup>-hour topology was executed in the day. Link available-bw is comparatively low since cross traffic in the day is more congested and busy than those during the night; RichMap did perceive the right situation.

Note that the two copies were different because some links were continuously added to the topology by the topology discovering thread. During the process of measuring a copy of the topology,  $R_0$  refreshes the available-bw of nearby links more frequently than the remote links. To smartly monitor a particular network area, the RichMap software provides an option that allows one to select a set of paths from the maps for timely measurements.

Now we study the effect of source node positions. Two RichMap instances were executed at two geographically different positions of GUCAS network: one connected to backbone (as shown in Fig. 8), the other stayed in a department LAN. These two locations are referred as center node and edge node, respectively. Fig. 9 shows that during the process of measuring a copy of the topology, how frequently a link will be traversed by probing packet trains, and how often a node must respond to the packet trains with

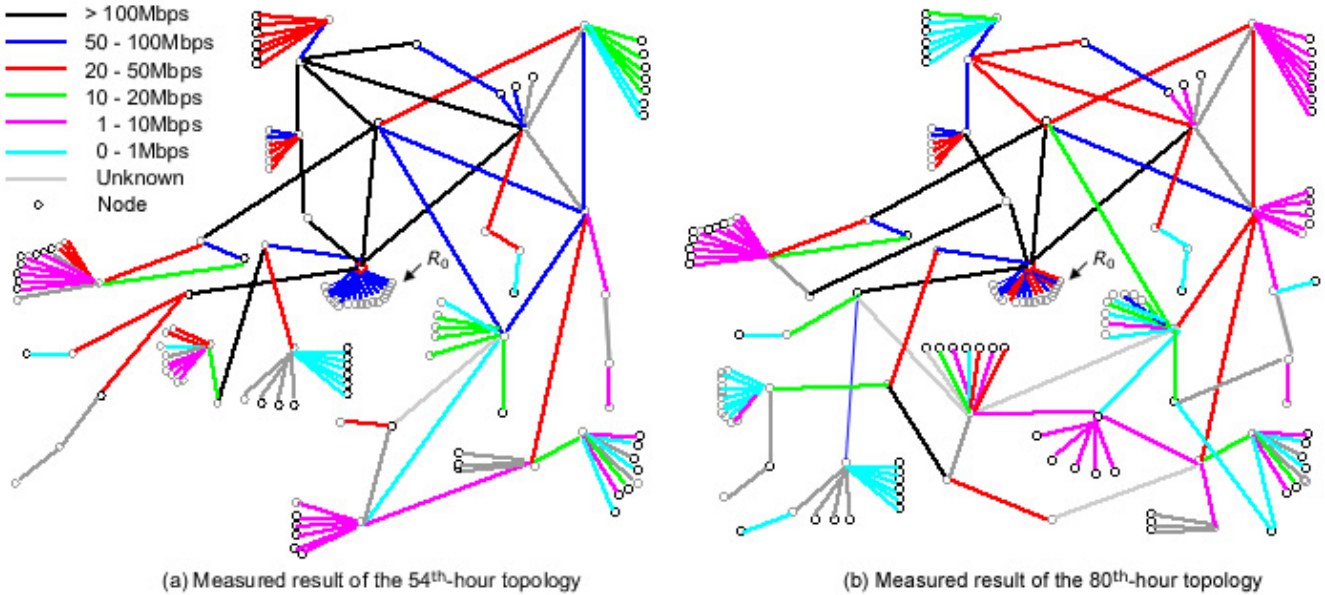
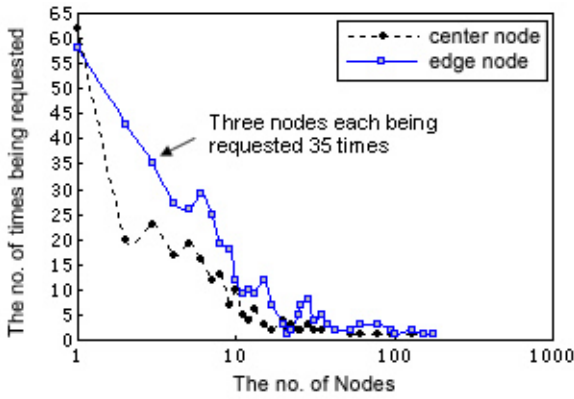
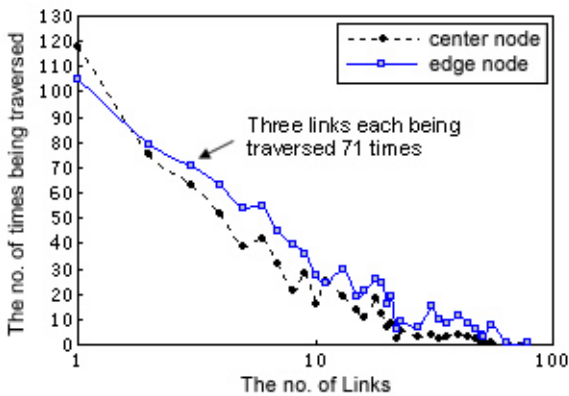


Fig. 8. Measured link available-bw on the topology of GUCAS network. We plot only the mean value of a rate range. If the upper bound of a rate range is unknown, then we plot the link available-bw as unknown.



(a) Node requested frequency



(b) Link traversed frequency

Fig. 9. Analysis of accessed nodes and links on GUCAS network.

ICMP packets. The result is that RichMap located at the edge node generally requires much more time and payload than that at the center node to cover the topology. In addition, since  $A_1$  of the edge node is generally lower than that of the center node, the edge node typically results in a map that cannot cover a wide range of rates as the center node does.

Finally, our later investigation revealed that RichMap tends to obtain accurate link available-bw when most of the nodes are no more than 6-hop away from  $R_0$ . Recall that RichMap first measures short paths of small hops, and then explores long paths of large hops (Section IV). As the hop

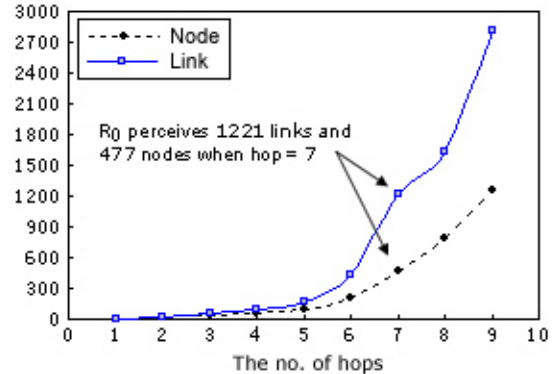


Fig. 10. The hop-limited scope of  $R_0$  on Beijing Telecom network.

number increases, all the links and nodes that RichMap must cover grows quickly (Fig. 10). Although 6-hop is enough for campus-link networks, it limits the scope of RichMap. Many factors contribute to this effect, e.g. the dynamic traffic conditions, and the unexpected router behavior. In the experiments, we found that these factors are correlative, and they step in together as RichMap attempts to measure long paths. Thus, an accurate scheme that can cover large-scale networks requires many more comprehensive considerations than we have made.

## VII. CONCLUSIONS AND FUTURE WORK

RichMap is designed to measure all link available-bw on the network topology from an arbitrary end node. The implementation of RichMap combines many heuristics from both topology discovery and available-bw measurement. Specifically, we have evaluated RichMap through simulations and networks experiments over 12 networks. The result shows that RichMap is able to accurately and efficiently capture the link available-bw of surrounding networks when most of the nodes are no more than six hops away from the source node. In addition, RichMap can also construct comparatively complete topologies of both small-scale and medium-scale networks.

This paper analyzed the approaches to topology discovery and available-bw measurement; many issues require further study, including the impact of dynamic router behavior, the modes of interaction between probing packet trains and diverse cross traffic, as well as the perception of networks from the perspective of an end node. We also hope to improve RichMap by adding more features to precisely identify PPD expansion in the presence of long paths, as well as to speed up the measurement process regardless of the source node position.

#### ACKNOWLEDGMENT

The authors would like to thank D. Towsley, J. Stankovic and D. Li for their constructive suggestions, Wasif, Nassar, Hu Wu, Mingjun Zhou, Dapeng Liu, Feng Yuan, Fengdi Shu, Xiaoyong Huai, Jinhui Zhou, Qing Wang, and Mingshu Li for their support.

#### REFERENCES

- [1] Y. Breitbart, M. Garofalakis, B. Jai, C. Martin, R. Rastogi, and A. Silberschatz, "Topology discovery in heterogeneous IP networks: the NetInventory system", *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, June 2004, pp. 401–414.
- [2] T. Bu and D. Towsley, "On distinguishing between Internet power law topology generators", in *Proc. IEEE INFOCOM*, New York, USA, June 2002.
- [3] H. Burch and B. Cheswick, "Mapping the Internet", *IEEE Computer*, vol. 32, no. 4, April 1999, pp. 97–98.
- [4] K. Calvert, M. B. Doar, and E. Zegura, "Modeling Internet topology", *IEEE Communications Magazine*, vol. 35, no. 6, June 1997, pp. 160–163.
- [5] R. Carter and M. Crovella, "Measuring bottleneck link speed in packet-switched networks", *Performance Evaluation*, vol. 27, no. 28, 1996, pp. 297–318.
- [6] J. Case, M. Fedor, M. Schoffstall, and J. Davin, "A simple network management protocol (SNMP)", *RFC 1157*, IETF, May 1990.
- [7] Q. Chen, H. Chang, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, "The origin of power laws in Internet topologies revisited", in *Proc. IEEE INFOCOM*, New York, June 2002.
- [8] K. Claffy and D. McRobb, "Measurement and visualization of Internet Connectivity and Performance", <http://www.caida.org/tools/skitter/>.
- [9] C. Dovrolis, P. Ramanathan, and D. Moore, "Packet dispersion techniques and a capacity estimation methodology", *IEEE/ACM Transactions on Networking*, vol. 12, December 2004, pp. 963–977.
- [10] A. B. Downey, "Using pathchar to estimate Internet link characteristics", in *Proc. ACM SIGCOMM*, Cambridge, USA, September 1999.
- [11] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the Internet topology", in *Proc. ACM SIGCOMM*, Cambridge, USA, September 1999.
- [12] V. Fuller, T. Li, J. Yu, and K. Varadhan, "Classless inter-domain routing (CIDR): an address assignment and aggregation strategy", *RFC 1519*, IETF, September 1993.
- [13] R. Govindan and H. Tangmunarunkit, "Heuristics for Internet map discovery", in *Proc. IEEE INFOCOM*, Tel Aviv, Israel March 2000.
- [14] N. Hu, L. Li, Z. M. Mao, P. Steenkiste, and J. Wang, "Locating Internet bottlenecks: algorithms, measurements, and implications", in *Proc. ACM SIGCOMM*, Portland, USA, August 2004.
- [15] M. Jain and C. Dovrolis, "Pathload: a measurement tool for end-to-end available bandwidth", in *Proc. Passive and Active Measurements (PAM) Workshop*, Collins, Colorado, USA, March 2002.
- [16] M. Jain and C. Dovrolis, "End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput", *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, August 2003, pp. 537–549.
- [17] R. Kapoor, L. Chen, L. Lao, M. Gerla, and M. Sanadidi, "CapProbe: A Simple and Accurate Capacity Estimation Technique", in *Proc. ACM SIGCOMM 2004*, Portland, USA, August 2004.
- [18] S. Keshav, "An Engineering Approach to Computer Networking", Addison-Wesley, 1997.
- [19] K. Lai and M. Baker, "Measuring link bandwidths using a deterministic model of packet delay", in *Proc. ACM SIGCOMM 2000*, Stockholm, Sweden, September 2000.
- [20] K. Lai and M. Baker, "Nettimer: a tool for measuring bottleneck link bandwidth", in *Proc. USENIX Symposium on Internet Technologies and Systems*, San Francisco, USA, March 2001.
- [21] Network Simulator (NS-2), <http://www.isi.edu/nsnam/ns/>.
- [22] J. Pansiot and D. Grad, "On routes and multicast trees in the Internet", *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 1, January 1998, pp. 41–50.
- [23] V. Paxson, "End-to-end routing behavior in the Internet", *IEEE/ACM Transactions on Networking*, vol. 5, no. 5, October 1997, pp. 601–615.
- [24] V. Paxson, "End-to-end Internet packet dynamics", *IEEE/ACM Transactions on Networking*, vol. 7, no. 3, June 1999, pp. 277–292.
- [25] J. Postel, "Internet control message protocol", *RFC 792*, IETF, 1981.
- [26] R. Siamwalla, R. Sharma, and S. Keshav, "Discovering Internet topology", July 1998, <http://www.cs.cornell.edu/skeshav/papers.html>.
- [27] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, "Measuring ISP topologies with Rocketfuel", *IEEE/ACM Transactions on Networking*, vol. 12, no. 1, February 2004, pp. 2–16.
- [28] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, "Network topology generators: degree-based vs. structural", in *Proc. ACM SIGCOMM*, Pittsburgh, USA, August 2002.
- [29] K. Thompson, G. J. Miller, and R. Wilder, "Wide-area Internet traffic patterns and characteristics", *IEEE Network*, vol. 11, no. 6, November 1997, pp. 10–23.
- [30] B. M. Waxman, "Routing of multipoint connections", *IEEE Journal of Selected Areas in Communications*, vol. 6, no. 9, December 1988, pp. 1617–1622.
- [31] E. Zegura, K. Calvert, and M. Donahoo, "A quantitative comparison of graph-based models for Internet topology", *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, December 1997, pp. 770–783.
- [32] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, "On the constancy of Internet path properties", in *Proc. ACM SIGCOMM Internet Measurement Workshop*, San Francisco, USA, November 2001.
- [33] H. Zhou, Y. Wang, and Q. Wang, "Measuring Internet bottlenecks: location, capacity, and available bandwidth", in *Proc. International Conference on Computer Networks and Mobile Computing (ICCNMC)*, Zhangjiajie, China, August 2005.
- [34] H. Zhou, Y. Wang, X. Wang, and X. Huai, "Difficulties in Estimating Available Bandwidth", in *Proc. IEEE International Conference on Communications*, Istanbul, Turkey, June 2006.



**Hui Zhou** received the B.S. degree in computer science from University of Science and Technology of China in 2002, the M.S. degree in computer sciences from Graduate University of Chinese Academy of Sciences (GUCAS) in 2005.

He is currently a PhD candidate in GUCAS, a project manager of National Science Foundation of China, and also the executive committee of YOCSEF-GS in China Computer Federation. He mainly focuses on the research of Internet measurement, network security, and P2P multimedia streaming.

Mr. Zhou received the Best Paper Award in the International Conference of Computer Network and Mobile Computing (ICCNMC) in 2005, the Microsoft Fellowship from Microsoft Research Asia in 2005.



**Yongji Wang** received the B.S. and M.S. degrees from Beijing University of Aeronautics and Astronautics in 1984 and 1987, respectively, and received the PhD degree from University of Edinburgh, United Kingdom in 1995.

He is currently a Professor in the Institute of Software, Chinese Academy of Sciences, and serves as a program committee for more than ten international journals and conferences.

Prof. Wang has long been engaged in the research of computer-controlled real-time systems, artificial intelligence, and computer network, and he was awarded the Hong Kong Wang Kuancheng Fund Award, British Overseas Scholars Fund Award, and Ford Fund Award.