Measurement Tool of One-Way Packet Loss Rates Based on Network Tomography

著者	Tsuru Masato, Ryoki Nobuo, Oie Yuji
journal or publication title	IEICE Transactions on Information and Systems
volume	86
number	11
page range	2334-2342
year	2003-11-01
URL	http://hdl.handle.net/10228/00006349

PAPER *Special Issue on New Technologies in the Internet and their Applications*

Measurement Tool of One-Way Packet Loss Rates Based on Network Tomography*

Masato TSURU[†], **Nobuo RYOKI**[†], ***and*** **Yuji OIE**[†], ***Regular Members***

SUMMARY The recent evolution on the network tomography have successfully provided principles and methodologies of inferring network-internal (local) characteristics solely from end-to-end measurements, which should be followed by deployment in practical use. In this paper, two kinds of user-oriented tools for inferring one-way packet losses based on the network tomography are proposed. They can infer one-way packet loss rates on paths or path segments from/to a user-host (a client) to/from a specified target host (an application server or a router) without any measurement on the target, and thus can find the congested area along the path between the client and an application server. One is a stand-alone tool running on the client, and the other is a client-server style tool running on both the client and a proxy measurement server distributed in the Internet. Prototypes of the tools have been developed and evaluated by experiments in the actual Internet environment, which shows that the tools can infer the loss rates within 1% errors in various network conditions.

key words: *measurement tool, one-way packet loss, network tomography*

1. Introduction

Since the Internet is characterized by its huge scale, diversity, and distributed administration, it is expensive, or sometimes difficult to directly measure internal dynamic states and performance of the network. Therefore, it is of practical importance to develop statistical and indirect ways to infer several network-internal (local) characteristics (e.g., packet loss rates and queuing delay statistics). Fortunately, the recent evolution on the network tomography have provided principles and methodologies of inferring network-internal characteristics solely from end-to-end measurements [1]–[3]. In particular, methods of inferring packet loss rates on a link or a network-cloud in paths has been successfully developed in recent studies [4]–[6], which employ closely spaced unicast packet pairs along tree-structured paths. Nevertheless, user-oriented (i.e., easily usable on end-hosts) measurement tools employing these methods have not been in practical use yet.

Measuring one-way characteristics of a path between a user-host and an application server is of practical importance for recent applications and application-level traffic optimizations, and it is sometimes required to be measured solely by the user-host. In this paper, therefore, we propose and develop two kinds of user-oriented tools for inferring one-way packet loss rates on path segments from/to an end-host (a client) to/from a specified target host (an application server or a router) without any measurement on the target. One is a stand-alone tool running on the client, and the other is a client-server style tool running on both the client and a proxy measurement server distributed in the Internet. They utilize the above-mentioned inference method based on the network tomography along tree-structured paths.

The remainder of this paper is organized as follows. Section 2 briefly explains how to infer packet loss rates on a path segment. Section 3 describes and discusses our prototype tools. Section 4 shows their evaluation by experiments in the Internet environment. Finally Sect. 5 concludes this work.

2. Inference of Packet Loss Rates

We present a brief explanation of inferring packet loss rates on a path segment (a portion of a path) from end-to-end measurements according to our previous work [6].

Let us consider tree-structured paths in Fig. 1. Let a and b be the path from node 0 to 1 via 3 and the path from 0 to 2 via 3, respectively. Path segments $0 \rightarrow 3$, $3 \rightarrow 1$, and $3 \rightarrow 2$ are regarded as virtual "links", which may be a network-cloud between an end node and an intermediate node. Each link is labeled by a set of paths including the link (i.e., a, b and ab). Suppose we dispatch a series of independent trials. In each trial, a

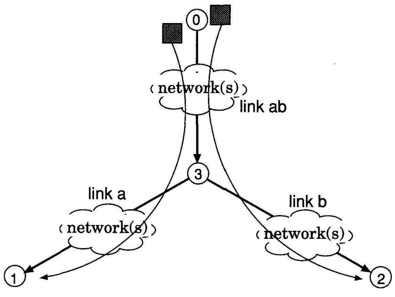


Fig. 1 End-to-end measurement on tree-structured paths.

Manuscript received March 1, 2003.
Manuscript revised May 31, 2003.

[†]The authers are with the Dept. of Computer Science and Electronics, Kyushu Institute of Technology, Iizuka-shi, 820–8502 Japan.

[*]A preliminary work was presented at SPIE ITCom 2002 [7].

pair of closely spaced (back-to-back) probe packets is sent from 0. The first packet P_a in a pair is bound for 1 along path a, while the second packet P_b is bound for 2 along path b. The time-space between P_a and P_b is as short as possible.

- Let N be the number of all trials, E_r ($r \in \{a, b\}$) be the number of trials in which P_r reaches the destination, and E_{ab} be the number of trials in which both P_a and P_b reach the destinations. We assume N is large enough.
- Let x_R ($R \in \{a, b, ab\}$) be the occurrence probability that an independent packet (entering link R) passes link R, which is one minus the packet loss rate on R. Note that x_{ab} can be regarded as the occurrence probability that the first packet P_a in a trial passes link ab, while the second packet P_b may be less likely to pass ab (i.e., more likely to be dropped on ab). In addition, let x^*_{ab} be the occurrence probability that both P_a and P_b in a trial pass link ab.
- Let y_r be the occurrence probability that P_r reaches the destination, and y_{ab} be the occurrence probability that both P_a and P_b in a trial reach the destinations. Note that y_a, y_b and y_{ab} can be simply estimated (by the sample means) from data N, E_a, E_b, and E_{ab} observed in end-to-end measurements.

Our goal is to infer packet loss rates on links a and ab (i.e., $1 - x_a$ and $1 - x_{ab}$), averaged in a measurement period (a period in which N-trials are performed) from only end-to-end measurements (i.e., N, E_a, E_b, and E_{ab}). To clarify the relation among the above probabilities, we express them by using four fundamental events:

$$\begin{aligned} &x_{ab} \overset{\text{def}}{=} \Pr[X^{(1)}_{ab}], \quad x^*_{ab} \overset{\text{def}}{=} \Pr[X^{(1)}_{ab} \cap X^{(2)}_{ab}], \\ &x_a \overset{\text{def}}{=} \Pr[X_a | X^{(1)}_{ab}], \quad x_b \overset{\text{def}}{=} \Pr[X_b | X^{(2)}_{ab}], \\ &y_a \overset{\text{def}}{=} \Pr[X^{(1)}_{ab} \cap X_a], \quad y_b \overset{\text{def}}{=} \Pr[X^{(2)}_{ab} \cap X_b], \\ &y_{ab} \overset{\text{def}}{=} \Pr[X^{(1)}_{ab} \cap X^{(2)}_{ab} \cap X_a \cap X_b] \end{aligned} \qquad (1)$$

where we let $X^{(1)}_{ab}$, $X^{(2)}_{ab}$, X_a, and X_b be the events of P_a passing link ab, P_b passing link ab, P_a passing link a, and P_b passing link b, in a trial, respectively.

Let us make some assumptions. Note that a typical situation in which the following assumptions hold is as follows: Losses on a link occur by an overflow of its queue; The overflow is caused by many independent, diverse traffic across the link; and queue is managed as FIFO.

(A1) Probe packets are not always dropped on each link:

$$x_a, x_b, x_{ab} > 0. \qquad (2)$$

(A2) Probe packets are dropped on each link independently, i.e., the following quantity δ is small:

$$\delta \overset{\text{def}}{=} \frac{\Pr[X_a \cap X_b | X^{(1)}_{ab} \cap X^{(2)}_{ab}]}{x_a x_b} - 1. \qquad (3)$$

(A3) In each trial, given that the second P_b that immediately succeeds P_a is not dropped on link ab, the first P_a is also unlikely to be dropped on the link, i.e., the following non-negative quantity ε is small:

$$\varepsilon \overset{\text{def}}{=} 1 - \Pr[X^{(1)}_{ab} | X^{(2)}_{ab}]. \qquad (4)$$

Then we have the following relation:

$$\begin{aligned} &y_a = x_{ab} x_a, \quad y_b = x^*_{ab} x_b / (1 - \varepsilon), \\ &y_{ab} = x^*_{ab} x_a x_b (1 + \delta). \end{aligned}$$

From the above assumptions and relation, we obtain approximations (inferred values) $\hat{x}_{ab}$ and $\hat{x}_a$ to x_{ab} and x_a, respectively, by letting $\varepsilon = 0$ and $\delta = 0$ as follows:

$$\hat{x}_{ab} \overset{\text{def}}{=} \frac{y_a y_b}{y_{ab}}, \quad \hat{x}_a \overset{\text{def}}{=} \frac{y_{ab}}{y_b} \qquad (5)$$

where the bias errors can be estimated as follows:

$$\begin{aligned} \frac{\hat{x}_a}{x_a} &= \frac{x_{ab}}{\hat{x}_{ab}} \\ &= (1 - \varepsilon)(1 + \delta). \end{aligned} \qquad (6)$$

On the other hand, we obtain approximations to y_{ab}, y_a, and y_{ab} from end-to-end measurements:

$$y_{ab} \approx \frac{E_{ab}}{N}, \quad y_a \approx \frac{E_a}{N}, \quad y_b \approx \frac{E_b}{N} \qquad (7)$$

where $\approx$ means that the right-hand quantity converges to the left-hand quantity if $N \to \infty$. Consequently, we have:

$$\hat{x}_{ab} \approx \hat{x}^{(N)}_{ab} \overset{\text{def}}{=} \frac{E_a E_b}{E_{ab} N}, \quad \hat{x}_a \approx \hat{x}^{(N)}_a \overset{\text{def}}{=} \frac{E_{ab}}{E_b}. \qquad (8)$$

where $\hat{x}^{(N)}_{ab}$ and $\hat{x}^{(N)}_a$ are employed as the inference for x_{ab} and x_a, respectively. While the convergence error depends on N, the bias error depends on δ and ε. Note that ε is expected to be upper-bounded as follows:

$$\varepsilon \le 1 - x_{ab} \le 1 - \max(y_a, y_b), \qquad (9)$$

which is derived from $\Pr[X^{(1)}_{ab} | X^{(2)}_{ab}] \ge \Pr[X^{(1)}_{ab}]$, and holds in many natural cases. There are, however, some situations that may increase ε up to $1 - x_{ab}$, e.g., a non-FIFO queue (e.g., RED) in a congested router on path segment ab.

3. The Prototype Tools

We describe our prototype tools – a stand-alone tool running on the client; and a client-server style tool running on both the client and proxy measurement server(s) that are assumed to be distributed in the Internet.

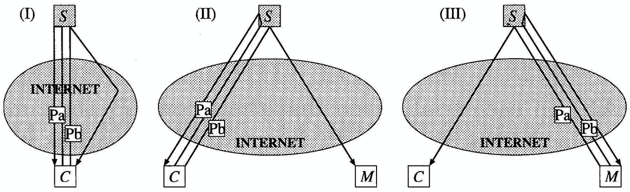


Fig. 2 Basic ideas of the inference tools.

A basic idea of the stand-alone tool is shown in (I) of Fig. 2, while that of the client-server style tool is shown in (II) and (III), where S is a target (a server or a router), C is a client who needs to infer the one-way losses between S and oneself, and M is a proxy measurement server. In general, a series of trials is dispatched where a pair (P_a, P_b) of closely spaced packets is sent in each trial.

For conciseness, let $\vec{CS}$, $\vec{SC}$, $\vec{SC}'$, $\vec{MS}$, and $\vec{SM}$ denote the one-way path (or path segment) from C to S, from S to C, from S to C via another roundabout route (in (I)), from M to S, and from S to M, respectively.

3.1 The Stand-Alone Tool

In (I) of Fig. 2, a series of trials is dispatched by C so as that P_a passes through $\vec{CS}$ and $\vec{SC}$, while P_b passes through $\vec{CS}$ and $\vec{SC}'$ in order to avoid the correlation between loss behaviors of P_a and P_b on the same backward path segment. This forms tree-structured paths, in which link ab and a in Fig. 1 correspond to $\vec{CS}$ and $\vec{SC}$, respectively. Based on the method explained in Sect. 2, the tool running on C infers one-way loss rate on each of $\vec{CS}$ and $\vec{SC}$.

To let the packets travel along the above tree, the most flexible way is to use the IP source route option, i.e., Loose Source and Record Route (LSRR), which is the very function of controlling the route of an IP packet by users. However, routers in the current Internet may prohibit packets with LSRR from being forwarded, because malicious use of LSRR can violate routing-policies of Internet Service Providers (ISP) and/or introduce a threat against security in the Internet. While the LSRR is not necessary for P_a if it causes S to return some reply packet (e.g., an ICMP message) regarded as a backward packet bound for C instead of P_a itself, P_b may need the LSRR option in general situations. A special case in which LSRR is not necessary is considered later (Fig. 3 (B)), which can be widely applicable in the current Internet.

3.2 The Client-Server Style Tool

In (II) of Fig. 2, a series of trials is dispatched by C so as that P_a passes through $\vec{CS}$ and $\vec{SC}$, while P_b passes through $\vec{CS}$ and $\vec{SM}$. On the other hand, in (III), in accordance with a request from C, M dispatches a series of trials so as that P_a passes through $\vec{MS}$ and $\vec{SC}$, while P_b passes through $\vec{MS}$ and $\vec{SM}$. Link ab in Fig. 1 corresponds to $\vec{CS}$ in (II) and $\vec{MS}$ in (III), respectively, while link a corresponds to $\vec{SC}$ in both types. In collaboration with M, the tool running on C infers one-way loss rates on each of $\vec{CS}$ and $\vec{SC}$ in (II), or on only $\vec{SC}$ in (III).

Note that, in (III), an upper-bound of ε (Eq. (9) of Sect. 2) implies that the fewer packet losses on $\vec{MS}$ ensures the more accurate inference of packet losses on $\vec{SC}$. Therefore, if there exist more than one proxy measurement servers, the most preferable server is the one showing the lowest round-trip losses to the target.

To let the packets travel along the above trees without use of the LSRR option, P_a and P_b should cause S to return some reply packets. Moreover, since the destination of the reply is usually determined by the IP source address in the received packet at S, C should send the second packet P_b by letting its IP source address be replaced with that of M in (II), and M should send the first packet P_a by letting its IP source address be replaced with that of C in (III).

In (II), unfortunately, since client C is likely to be on a user-host, a packet sent by C bound for the Internet with an IP source address not owned by C's local domain may be prohibited by an exit router of the domain or an edge router of the connected ISP from being forwarded. This is because a number of security incidents (e.g., DoS attack) caused by malicious use of modified IP source addresses have been reported. In (III), on the other hand, since proxy measurement server M is expected to be installed for a special and dedicated purpose (e.g., M may be installed by the ISP itself as a service function), the above problem can be avoided.

Furthermore, in (III), even though a firewall in front of C may prohibit any packets bound for C except for a reply to some packet sent from C before, to send a dummy (fake) packet from C to S (through the firewall) in advance avoids the problem in most cases.

Consequently, while type (II) is more efficient because one-way loss rates of the forward and backward directions are inferred simultaneously, type (III) is more widely applicable in the current Internet.

3.3 Prototype Implementations of the Type (I) and (III) Tools

We have developed prototypes of type (I) and type (III) tools. P_a and P_b are implemented by the UDP packets, and the replies to them are the UDP echo response or some ICMP message packets. Although our tools can use several possible ICMP messages, use of the Time-exceeded message seems suitable in many cases, because – i) most of the current routers properly reply to any time-exceeded packets (though a few routers do

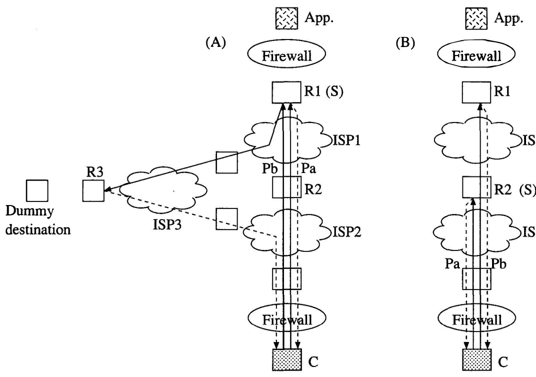


Fig. 3 Typical usage of the stand-alone tool.

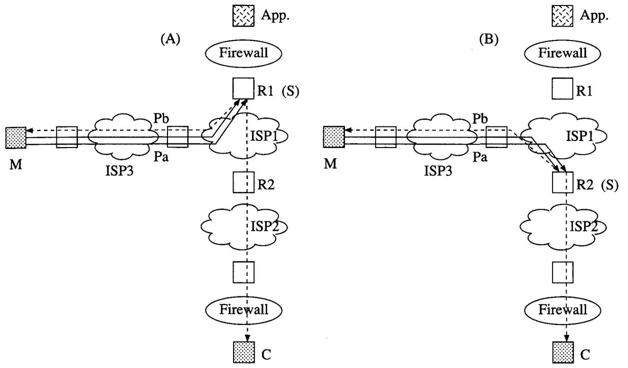


Fig. 4 Typical usage of the type (III) client-server style tool.

not reply to them); ii) they are unlikely to be suspected as malicious activities; and iii) they can sometimes detect the change of the route.

In order to infer the loss rate on a path to/from an application server, it is practical to infer that on a path segment to/from a boundary (entrance/exit) router in front of the application server instead (e.g., Fig. 3 (A) and Fig. 4 (A)), because – i) in most cases, the server is protected by a firewall, which may prohibit any measurement packets from reaching the server; ii) even if not, measurement packets should not affect the busy, heavy-loaded server; and iii) packet losses on the server's LAN (between the server and the boundary router) may be negligible. Note that an idea to infer one-way packet losses on the exact paths from/to a client to/from an application server is to utilize TCP protocol sequences under the application protocol itself (e.g., sting [8]). To extend our tools so as to use probing packets of TCP instead of UDP remains as future work.

In order to infer the loss rate on a path segment to/from a router residing in a path, it is assumed that – i) the route of the path (between the client and the application server), e.g., the router's IP address or the hop count along the path, is known a priori (e.g., by using traceroute); and ii) the route does not change within the measurement period. Note that "running traceroute on the client" is not a perfect solution to know (infer) the routes of the paths when the forward and backward paths have different routes, because it needs the IP source route option to infer the route of the backward path, which is sometimes prohibited as mentioned before. While several works have been trying to know the backward-route from a server, this remains for future studies.

In the above sense, our tools can infer one-way packet loss rates not only on a path between the client and an application server, but on a path segment of the path, and thus can find the congested area along the path. Figure 3 and Fig. 4 show detailed and realistic examples.

In Fig. 3, (A) and (B) show the routes of probe packets (the UDP packets with an appropriate Time-To-Live (TTL) and the ICMP Time-exceeded message packets as replies to them) when the stand-alone tool infers packet losses on both of the forward and the backward path segments between C and R_1 (a boundary router for S), and between C and R_2 (an intermediate router along the paths), respectively. In case (A), C sends P_b bound for a dummy destination with the LSRR option to be forwarded by way of R_1, and with some TTL value so that it will decrease to 0 at R_3 on a path to the dummy destination host. On the other hand, in case (B), C sends P_b simply bound for S with some TTL value so that it will decrease to 0 at R_1 without use of LSRR. In general, if there exists a router (R_1) beyond the target (R_2) along a path, then we can use R_1 as a reflector for the second packet P_b. A requirement from the inference method is that the two reply packets returned back to C are independent with respect to loss behaviors so that δ (Eq. (3) in Sect. 2) is small. Therefore, if the replies from R_1 and from R_2 passes through the same backward path segment in ISP2 so closely or a congestion on that segment continues so long, then one of the pair packets is likely to be dropped when the other is dropped, and thus, a significant inference (bias) error may be introduced in case (B).

Similarly, in Fig. 4, (A) and (B) show the routes of probe packets when the client-server style tool (type (III) in Fig. 2) infers packet losses on the backward path segments from R_1 to C, and from R_2 to C, respectively.

4. Experiments

4.1 Estimation of the Accuracy and Stability of the Method on a Test-Bed over the Internet

First we evaluated the inference accuracy and stability of the method on which our tools relied. This was done by comparing the inferred loss rates to the actual loss rates on a test-bed consisting of four nodes (UNIX PCs) over the Internet. These four PCs were distributed in the Internet of Japan, and we regarded one-way paths between them as links ab, a, and b in Fig. 1, by which we constructed six patterns of tree-structured paths. These paths traversed four major ISPs and two major Internet exchanges (IX) in Japan. We have performed a number of experiments running the type (III) tool under various parameters in different day, time and path configurations, over three months.

On the test-bed, since probe packets could be observed at the intermediate node (node 3 in Fig. 1), the actual loss rates of the probe packets on each segment were observable. Let I_r ($r \in \{a, b\}$) be the number of trials in which P_r reaches the internal node 3, and I_{ab} be the number of trials in which both P_a and P_b reach node 3. Then, we have:

$$x_{ab} \approx \frac{I_a}{N}, \quad x_a \approx \frac{E_a}{I_a}, \qquad (10)$$

$$\delta \approx \frac{E_{ab} I_a I_b}{I_{ab} E_a E_b} - 1, \quad \varepsilon \approx 1 - \frac{I_{ab}}{I_b} \qquad (11)$$

where E_a, E_b, and E_{ab} are defined in Sect. 2. Recall ε and δ are the two main factors of bias errors in the inference as described in Eq. (6) in Sect. 2. The right-hand quantities can be regarded as the actual values though they include the convergence errors.

Table 1 shows the tool parameters we examined. Note that we did not see particular differences in inference accuracy among two packet types and three packet sizes in our experiments. There are some conflicting factors: i) The larger N can lead to the smaller convergence errors, which is vital to the loss inference for path segments whose conditions are fairly good and loss rates are low; ii) The smaller N or the shorter Int (the mean inter-trial time) can lead to the shorter measurement period, which is preferable to capture the change of states (time-varying characteristics); and iii) The longer Int may lead to the smaller side-effect to the network (i.e., less traffic load) and the lower correlation among trials, which depend on the network states and bandwidths on the paths. In our experiments, $N = 3000$ and $Int = 0.1$ (sec) seemed to be an appropriate choice, which implied that what we infer was the 5-minutes mean packet loss rates. The reasons for this are as follows: 1) In most of our experiments, loss rates were less than 0.01 (1%) and should be investigated in the order below 0.5%. Thus, since the number of lost packets for capturing such losses is no more than $N \times 0.005$, the cases with $N = 1000$ or 2000 cannot be expected to produce reliable statistics for those losses; 2) According to a study on the constancy of Internet path properties [9], the loss rates are likely to stable in a few or ten minutes. In this sense, it may be reasonable that $Int \times N$ must be less than 10 minutes, and thus, combined with $N = 3000$, both of $Int = 0.1$ and 0.05 ($\sim$ 5 and 2.5 mins.) are acceptable; 3) In our experiments, we found the results in cases with $Int = 0.1$ and 0.05 were quite similar, and thus, $Int = 0.1$ is preferable because of its less side-effect.

On the other hand, since the measurement performed by our tools may resemble DoS attacks, the smaller N and the larger Int are preferable at the security point of view. To the end, the optimal parameters may strongly depend on the requested accuracy, requirements from the environment, and the network condition. A systematic optimization of those parameters is of practical importance and remains as future work.

Figure 5 shows the actual loss rates on links ab and a (by $1 - x_{ab}$ and $1 - x_a$ using the right-hand sides in Eq. (10)) and inferred loss rates of them (by $1 - \hat{x}_{ab}$ and $1 - \hat{x}_a$ using the right-hand sides in Eq. (8) in Sect. 2) being calculated in 3000 successive trials over all experiments using random inter-trial time (uniform and exponential distributions), which indicates the accurate inference through a broad range of loss rates.

Figure 6 shows the time-variation of the actual and inferred loss rates on links ab and a (i.e., $1 - x_{ab}$, $1 - \hat{x}_{ab}$, $1 - x_a$, and $1 - \hat{x}_a$) in two sample experiments, where each experiment consisted of 10000 successive trials with exponentially distributed inter-trial time (0.1 sec mean). The actual and inferred loss rates were calculated in each moving measurement window consisting of 3000 successive trials shifted by 500 trials (i.e., [1, 3000], [501, 3500], …, [7001, 10000]). The

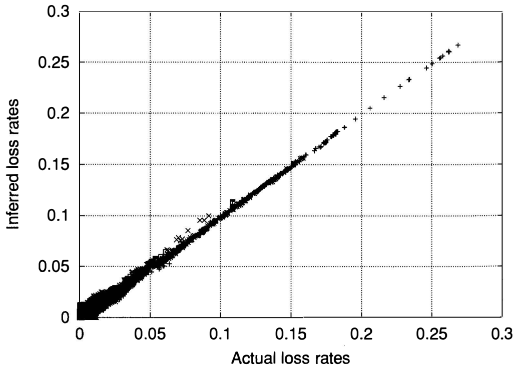


Fig. 5 Comparison between actual and inferred loss rates.

Table 1 Measurement parameters.

Parameters	Tested values
Probe packet type	UDP-echo, UDP+ICMP
UDP packet size	64, 256, 1400 (bytes)
Inter-trial time distribution	fixed, uniform, exponential
Int (mean inter-trial time)	0.1, 0.05 (sec)
N (# of trials in a measurement period)	1000, 2000, 3000

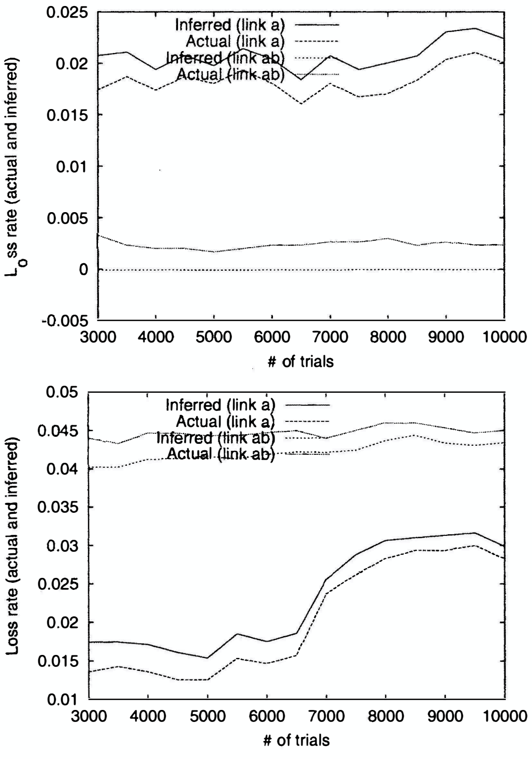


Fig. 6 Actual and inferred loss rates in two sample experiments.

inferred values seem to well track actual loss rates.

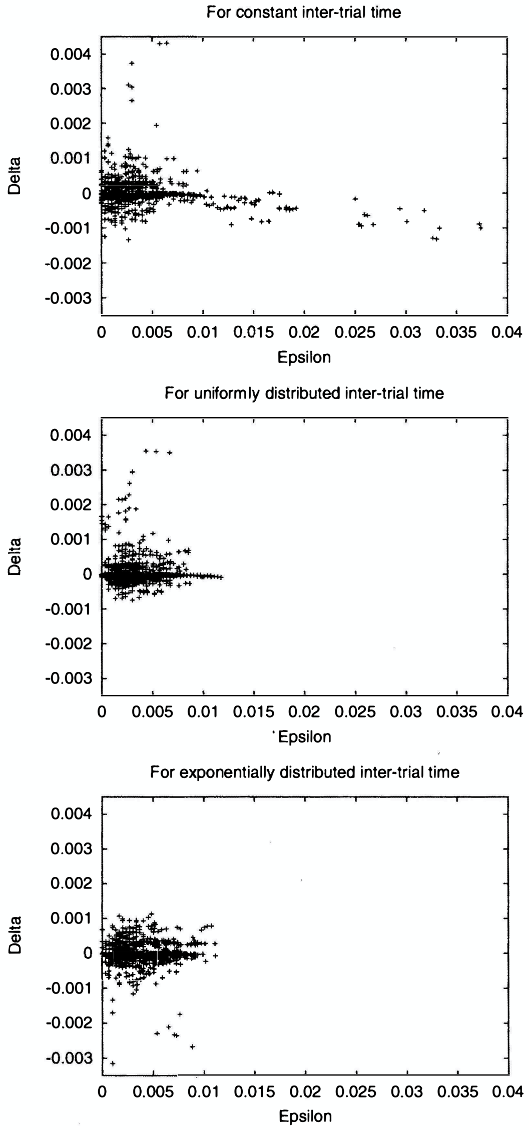
Figure 7 shows (ε, δ)-plot over the experiments with each specific inter-trial distribution of fixed, uniform, and exponential distributions (each of them consists of about 2000 experiments) where ε and δ were calculated in 3000 successive trials by Eq. (11). Since the bias error depends on ε and δ linearly (Eq. (6) in Sect. 2), ε was dominant, while δ was negligible, with respect to the inference accuracy in our experiments. Note that, from Eq. (6), a large ε causes an over-estimation of x_{ab} (i.e., an under-estimation of the loss rate on link ab) and an under-estimation of x_a (i.e., an over-estimation of the loss rate on link a), which agrees with Fig. 6. In the worst cases, ε was close to 0.04, which means 4% errors in $\hat{x}_{ab}$ and $\hat{x}_a$. However, such inaccuracy arose only in cases with a fixed inter-trial time. In cases with randomly (i.e., uniformly or exponentially) distributed inter-trial time, ε was less than or equal to 0.01, i.e., 1% errors in $\hat{x}_{ab}$ and $\hat{x}_a$, which verified the acceptable accuracy and stability of the inference method.

Figure 8 verifies the property of the upper-bound of ε (Eq. (9) in Sect. 2).

4.2 Validation of the Tools in Actual Environments on the Internet

Next we tried our tools in practical use, that is, inferring one-way packet loss rates from several routers (in front of some famous web servers) in the Internet to one of our nodes. Since the actual loss rates were not available for us in such environments, we validated the inference performed by our tools in two indirect ways. One was to compare the inferred values by two instances of the type (III) client-server style tool running simultaneously. The other was to compare the inferred values by an instance of the type (III) client-server style tool and two instances of the type (I) stand-alone tool running simultaneously. We employ the following parameters in all measurements: a 64 bytes UDP packet and its reply of ICMP Time-exceeded message, exponentially distributed inter-trial time with 0.1 sec mean, and $N = 3000$.

Figure 9 shows the idea of the former experiment. We performed two instances of the client-server style tool at the same time; they inferred the same one-way


For constant inter-trial time

For uniformly distributed inter-trial time

For exponentially distributed inter-trial time

Fig. 7 (ε, δ)-plot over the experiments with each specific inter-trial distribution.

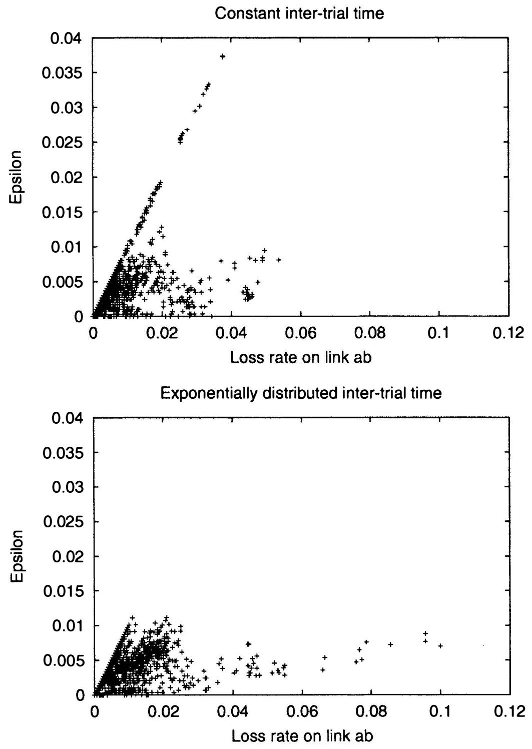


Fig. 8 Loss rates on link *ab* v.s. ε for two types of inter-trial distributions (top: fixed, bottom: exponential).

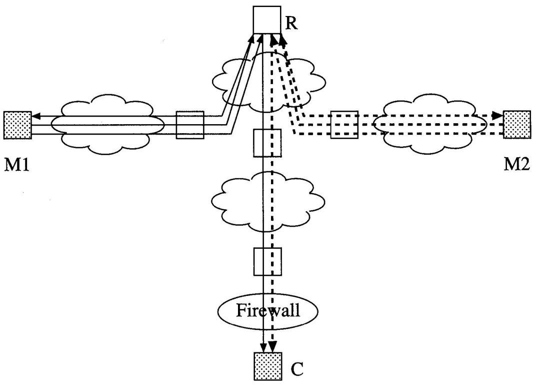


Fig. 9 Validation of the type (III) tool in actual environments.

Table 2 The hops from proxy measurement servers and a client to targets in Fig. 9.

R (the target)	M_1	M_2	C
210.81.90.190	12	13	20
210.130.160.180	10	13	19
210.80.37.65	12	13	21
150.100.1.130	7	9	16

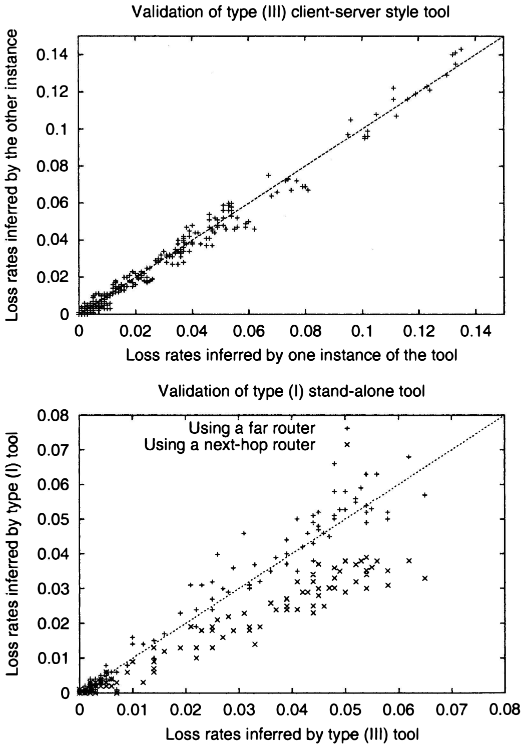


Fig. 10 Comparison among the loss rates inferred by the different instances of the tools.

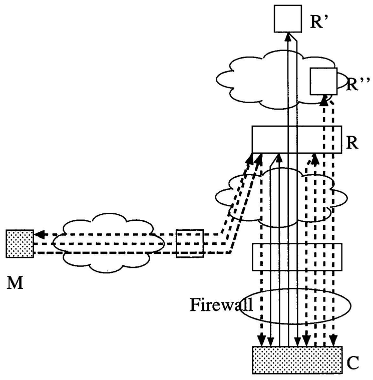


Fig. 11 Validation of the type (I) tool in actual environments.

loss rates from router R to C where we employed four different routers in the Internet as R. One of two instances run on C and M_1, while the other run on C and M_2. Table 2 shows the distances (hops) from M_1, M_2 and C to the four target routers (as R).

The top one in Fig. 10 compares inferred loss rates by them in a number of experiments. The differences between each pair of inferred values are less than or equal to 0.01, which are consistent with the previous result of 1%-accuracy.

Figure 11 shows the idea of the latter experiment. We performed one instance of the client-server style tool (that is expected to give good approximations of actual loss rates), and two instances of the stand-alone tool without use of LSRR (that should be examined on the accuracy of inferring loss rates) at the same time; they inferred the same one-way loss rates from router R to C. Both of the stand-alone tool instances running on C sent the second packet P_b to an application server; one of them sent P_b so as to cause router R' (7-hops far beyond R) to return a reply, while the other in-

Table 3 The route from a client to an application server in Fig. 11.

hop	address	rtt1 (ms)	rtt2 (ms)	rtt3 (ms)
1	131.206.38.254	0.321	0.304	0.263
		......		
12	150.99.197.65	142.347	147.357	140.077
13	150.99.197.5	144.161	145.694	*
14^R	150.99.197.26	162.191	180.358	*
$15^{R''}$	150.99.99.28	165.938	177.209	183.332
16	62.40.103.197	435.363	443.906	433.179
		......		
20	146.97.35.86	438.847	427.946	440.030
$21^{R'}$	146.97.40.82	442.338	448.333	445.698
22	163.1.0.90	426.015	528.339	428.483

stance sent P_b so as to cause router R'' (next to R) to return a reply. Table 3 shows an output of the "traceroute" from client C to the application server where we employed 150.99.197.26 as R, 150.99.99.28 as R', and 146.97.40.82 as R''.

The bottom one in Fig. 10 compares inferred loss rates by them in a number of experiments. While the stand-alone tool using a router (as a reflector for the second packet) far beyond the target router showed the similar inference accuracy as the client-server style tool, that using a next-hop router exhibited more inference errors (over-estimations of x_a), which must be due to some degree of δ. This result implies possible cases in which the stand-alone tool can work accurately: (i) there exists an appropriate router (as a reflector for the second packet P_b) far beyond the target router; (ii) the degree of δ (due to P_a and P_b closely traveling) can be estimated in some way; or (iii) the LSRR option is available for letting packets travel along appropriate tree-structured paths.

5. Concluding Remarks

We have developed prototype tools for inferring one-way packet loss rates on a path segment from/to an end-host (a client) to/from a specified target host (an application server or a router) without any measurement on the target. We have evaluated them by experiments in the Internet environment, which showed that the tools could infer the loss rates within 1% errors in various network conditions. For improvement and deployment of our tools, we need to conduct more experiments and analysis in various actual environments. In particular, it should be studied how to decide a set of appropriate parameters and how to reliably estimate the degree of the inference error.

We are also trying to extend the tools in order to infer queuing delay statistics [10], [11] and bottleneck bandwidth [12] on a path segment to/from a target host.

Acknowledgments

This work was mainly done while the first author was with the Telecommunications Advancement Organization of Japan (TAO). For the tool development and distributed experiments, we must thank many people: Fumito Kubota, Shinichi Nakagawa, Syuji Uetsuki, and Yoshiaki Kitaguchi at Communications Research Laboratory, Ryuji Kawakita, Mina Hirowatari, Akiko Isobe, and Masashi Ueno at Japan Telecom Information Service Co., Ltd., and Takeshi Ikenaga, Takuya Nakashima, Kaori Hirono, and Seiji Kaneko at Kyushu Institute of Technology.

References

[1] A. Adams, T. Bu, T. Friedman, J. Horowitz, D. Towsley, R. Caceres, N. Duffield, F.L. Presti, S.B. Moon, and V. Paxson, "The use of end-to-end multicast measurement for characterizing internal network behavior," IEEE Commun. Mag., vol.38, no.5, pp.152–158, May 2000.

[2] M. Coates, A. Hero, R. Nowak, and B. Yu, "Internet tomography," IEEE Signal Process. Mag., vol.19, no.3, pp.47–65, May 2002.

[3] M. Tsuru and Y. Oie, "Introduction to the network tomography," IEICE Technical Report, IN2001-106, Nov. 2001.

[4] M. Coates and R. Nowak, "Network loss inference using unicast end-to-end measurement," Proc. ITC Conf. IP Traffic, Modeling and Management, pp.28-1–28-9, Monterey, Sept. 2000.

[5] N. Duffield, F.L. Presti, V. Paxson, and D. Towsley, "Inferring link loss using striped unicast probes," Proc. IEEE infocom, Anchorage, April 2001.

[6] M. Tsuru, T. Takine, and Y. Oie, "Inferring link loss rates from unicast-based end-to-end measurement," IEICE Trans. Commun., vol.E85-B, no.1, pp.70–78, Jan. 2002.

[7] M. Tsuru, N. Ryoki, T. Nakashima, and Y. Oie, "A measurement tool of one-way characteristics based on network tomography," Proc. SPIE ITcom, vol.4865, pp.98–107, Boston, July 2002.

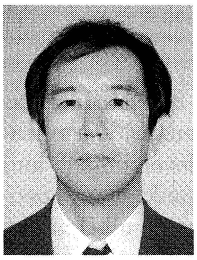
[8] S. Savage, "Sting: A TCP-based network measurement tool," Proc. USENIX Symposium on Internet Technologies and Systems, 1999.

[9] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, "On the constancy of Internet path properties," Proc. ACM SIGCOMM Internet Measurement Workshop, San Francisco, Nov. 2001.

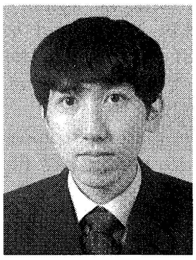
[10] M. Coates and R. Nowak, "Network delay distribution inference from end-to-end unicast measurement," Proc. IEEE Int. Conf. Acoust., Speech and Signal Process., Salt Lake City, May 2001.

[11] N. Duffield, J. Horowitz, F.L. Presti, and D. Towsley, "Network delay tomography from end-to-end unicast measurements," Proc. Int. Workshop on Digital Communications, Taormina, Italy, Sept. 2001.

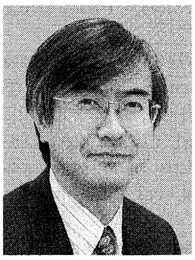
[12] K. Harfoush, A. Bestavros, and J. Byers, "Measuring bottleneck bandwidth of targeted path segments," Proc. IEEE infocom, San Francisco, April 2003.



Masato Tsuru received B.E. and M.E. degrees from Kyoto University, Japan in 1983 and 1985, respectively, and then received his D.E. degree from Kyushu Institute of Technology, Japan in 2002. He worked at Oki Electric Industry Co., Ltd. (1985–1990), Information Science Center, Nagasaki University (1990–2000), and Japan Telecom Information Service Co., Ltd./ Telecommunications Advancement Organization of Japan (2000–2003). Since April 2003, he has been an Associate Professor in the Department of Computer Science and Electronics, Kyushu Institute of Technology. His research interests include performance measurement, modeling and analysis of computer communication networks. He is a member of the IPSJ and JSSST.



Nobuo Ryoki received the B.E. and M.E. degree in information engineering from Kyushu Institute of Technology, Iizuka, Japan, in 1999 and 2001, respectively. Since 2001, he has been a Ph.D. candidate at the Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology. His research interests include performance measurement and estimation techniques on the quality-aware internet and its applications. He is a member of the WIDE Project of Japan.



Yuji Oie received B.E., M.E. and D.E. degrees from Kyoto University, Kyoto, Japan in 1978, 1980 and 1987, respectively. From 1980 to 1983, he worked at Nippon Denso Company Ltd., Kariya. From 1983 to 1990, he was with the Department of Electrical Engineering, Sasebo College of Technology, Sasebo. From 1990 to 1995, he was an Associate Professor in the Department of Computer Science and Electronics, Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology, Iizuka. From 1995 to 1997, he was a Professor in the Information Technology Center, Nara Institute of Science and Technology. Since April 1997, he has been a Professor in the Department of Computer Science and Electronics, Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology. His research interests include performance evaluation of computer communication networks, high speed networks, and queuing systems. He is a member of the IEEE and IPSJ.