IP Performance Management Infrastructure for ISP

Atsuo Tachibana, Yuichiro Hei, Tomohiko Ogishi, and Shigehiro Ano

KDDI R&D Laboratories, 2-1-15 Fujimino-shi, Saitama, Japan
{tachi,hei,ogishi,ano}@kddilabs.jp

Abstract. An IP performance management infrastructure that has the twin frameworks of performance measurement and topology monitoring is proposed. By combining above frameworks, the infrastructure locates performance-degraded segments. Since the Internet is still highly prone to performance deterioration due to congestion, router failure, and so on, not only detecting performance deterioration, but also monitoring topology and locating the performance-degraded segments in real-time is vital to ensure that Internet Service Providers can mitigate or prevent such performance deterioration. The infrastructure is implemented and evaluated through a real-world experiment and its considerable potential for practical network operations is demonstrated.

1 Introduction

Today's Internet serves as a communication infrastructure that supports various social and economic activities. The Internet needs to be managed in a reliable and efficient manner, and should be measurable and tractable in terms of its characteristics and internal states by ISPs (Internet Service Providers). However, it is widely known that transient performance deterioration is actually still likely to occur due to the inherent feature of providing cost-efficient and scalable services based on statistical, loosely controlled shares of network resources, meaning that ISP operators have to detect performance deterioration and take certain action to mitigate it.

Although SNMP (Simple Network Management Protocol) may provide high accuracy for a metric such as a link-by-link packet loss, it is not suitable for estimating customers' traffic performance. Hence, active measurement of packet behaviors passing through the targeted intra-network is an essential and rational approach to the network operations of ISPs. In addition to performance measurement, topology monitoring is important for network operation, because failures of a router or a transit network resulting in topology change can potentially severely degrade the performance of the network. Monitoring LSAs (Link-State Advertisements) that are updates of OSPF, a widely deployed intra-domain routing protocol, is a promising approach for ISP operators to efficiently monitor the topologies on the intra-network [1–3].

A variety of infrastructures and tools for active measurement and topology monitoring have been extensively developed for research (e.g., [4–11]). However,

because most of them aim to clarify some research issues based on a different network model and not designed for real-time network operation, their practical application to a commercial ISP's network operation in their present form remains difficult. Therefore in this paper, we propose an IP performance management infrastructure that has the twin frameworks of performance measurement and topology monitoring. In addition, there is a real possibility that the analysis of the combination of these two functionalities might reveal other useful features from the measurement data.

Locating performance-degraded segment ensures that ISP operators take action to reduce costs and configuration efforts required for troubleshooting. Network tomography is a promising analysis technique for inferring the internal status of a network solely by measuring end-to-end packet behaviors passing through multiple paths traversing the targeted intra-network instead of directly monitoring each network node such as routers. A variety of network tomographic approaches have been extensively studied (e.g., [12–16]). Along these lines, as a practical way to locate performance-degraded segments, the inference methods in real-time are also proposed [17, 18]. These methods actively and continuously measure end-to-end packet loss rate or delay variation within each measurement period (e.g., 15 seconds and 5 seconds, respectively) on paths from multiple origins to multiple destinations. The infrastructure includes these methods targeting ISPs' network operation.

The rest of the paper is organized as follows. Section 2 shows the system requirements. In Section 3, we explain the details of the system architecture, and in Section 4, in order to assess the feasibility of the infrastructure we outline an experiment with its results including the considerable potential of the proposed infrastructure for practical operational use. Finally, we conclude this work in Section 5.

2 System Requirements

Minimum system requirements for the proposed infrastructure are listed below.

1. Performance measurement
   First, network probes themselves are unsolicited and may cause harm to the network especially on a large scale, by consuming precious router processing resources. Probing packets have to be sufficiently light even when active measurements are simultaneously performed on multiple paths. To address this, we designed the following system requirements.
   - Probing packets are sent at a low rate (e.g., several Kbps) on each monitored path. To avoid being synchronized with a particular network-internal queue behavior, the interval of an adjacent probing packet should be chosen randomly as specified in the IPPM (IP Performance Metrics) Working Group of the IETF.

2. Topology monitoring
   Although traceroute command is one available approach to making route measurements, since traceroute lists the source addresses of“ Time exceeded ” ICMP messages, these addresses represent interfaces that received traceroute probes, the topology information obtained by traceroute is not necessarily accurate. In addition, to monitor all routers' interface states in a large-scale network precisely, a large number of traceroute probes would be required. Hence, this approach is not suitable for continuous topology monitoring for an entire large-scale network. Instead, in this paper, we focus on monitoring LSAs. Routers running OSPF advertise their link states on LSAs to the network, and OSPF routers can construct a complete view of the topology of the network from these LSAs. Therefore, by passively monitoring LSAs flooded throughout the intra-network, we can comprehend the topology of the network.

3. Combination of performance measurement and topology monitoring
   From the operator's viewpoint, not only IP performance on the measurement paths among measurement nodes but also the routes the probing packets pass through are important. For example, when performance deterioration is detected, operators investigate MIB (Management Information Base) of routers which are included in the monitored paths. Hence, the route of each monitored path needs to be calculated in real-time by combining the performance measurement information and topology.

4. Security
   In addition to the harmful aspect of network probing, network measurement data may be abused and used to harm the network. This is due to the fact that accurate measurement data may help unscrupulous parties to attack the infrastructure. Furthermore, network measurement data can expose private information on network architectures. Thus, the measurement infrastructure has to be securely and robustly managed including handling of measurement data.

5. Cost
   Although hardware-based measurement infrastructure may provide precise timestamps by using a specialized device with a built-in clock, it is difficult to apply to a large-scale distributed measurement infrastructure due to high cost and lack of expansibility. Hence, measurement nodes should be implemented with light-weight software. Our measurement software works even on a small PCs on which an RISC CPU 400-MHz processor, 64-MB RAM, 16-MB flash Rom, and two 100-Mbit Ethernet interfaces are implemented.

3 System Architecture

Our infrastructure consists of three frameworks for performance measurement, topology monitoring (i.e., OSPF monitoring), and analyzing measurement results as illustrated in Fig. 1.

The performance measurement framework continuously measures network performance along multiple paths among distributed measurement nodes. Measurement results such as one-way packet delay and loss rate along each path are periodically (e.g., every 1 minute) collected and stored in the performance database.

The OSPF monitoring framework captures LSAs at a measurement node that establishes an adjacent relationship with an OSPF running router. By analyzing LSAs, we build an entire view of the network topology and detect route instability. The topology database stores an entire network topology.

The data analysis framework detects performance deterioration by periodically looking up the performance database. Although network performance deterioration itself might be defined in various ways, we simply assume that packet loss rate or packet delay variation experienced on at least one monitored path is distinguishably larger than that found under normal conditions. Furthermore, the infrastructure infers the location of the performance-degraded segments by analyzing the performance database with the topology database along the line of network tomography.

Operators interact with all frameworks and databases through the web-based graphical interface that uses HTML4, Javascript, PHP4, and Apache Web server using SSL (Secure Sockets Layer). In the following subsections, we explain the details of each framework.

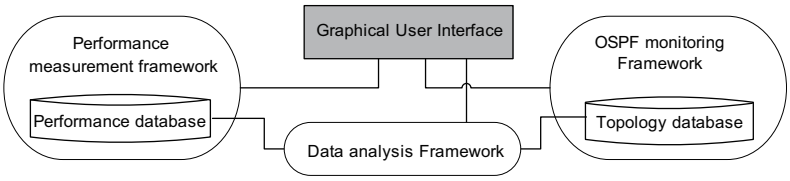

Fig. 1. System architecture

3.1 Performance Measurement Framework

The system components of the performance measurement framework are depicted in Fig. 2. The measurement management software is in charge of measurement control, comprising not only the execution of performance measurements, but also monitoring survivability and configuration of measurement nodes. Along multiple paths among measurement nodes, performance is actively measured and the results are uploaded to the performance database.

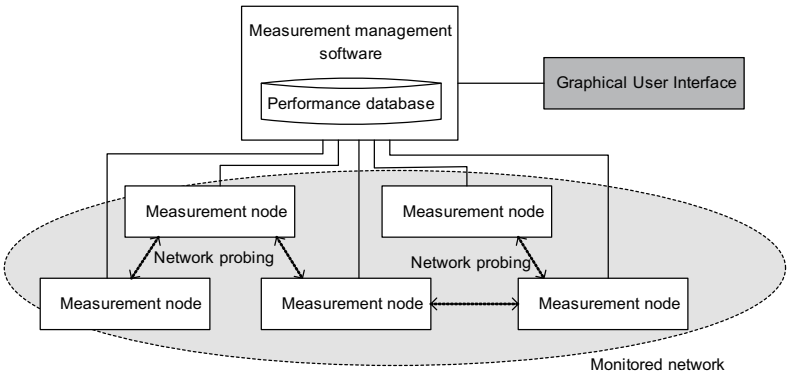

Fig. 2. The system components of the measurement framework

(1) Measurement node software

The measurement node software is implemented over the GNU Linux operating system. Measurement nodes are always listening for commands from the measurement management software. The functionality of the measurement node should be reduced to a minimum in order to improve the robustness of the whole platform in case of a single node failure. Accordingly, the node software only provides the following functions.

– Measuring IP performance by sending probing packets
UDP packets are sent at certain intervals (e.g., Poisson interval, uniformly distributed interval, and Pareto-distributed interval). The probing packet size is set by operators in the range from 64 to 1500 bytes. The TOS (Type of Service) field in the IP packet header is also specified for measurements over the policy-routing operated network.

– Computing IP performance metrics
The nodes periodically (e.g., every one minute) calculate 11 metrics at each period, the maximum, minimum, average, and the 95th percentile of one-way packet delay and delay variation respectively, the number of lost packets, loss rate, and the number of reordered packets. Here, we define packet delay variation v_k for k-th packet along a path as the difference between the absolute one-way delay x_k of the packet and the minimum d in all the observed absolute one-way delays along the path ($v_k = x_k - d$), which is nearly equal to the sum of queuing delays experienced by the packet along the path.

– Uploading the measurement results to the data server
The 11 metrics described above are periodically uploaded from each measurement node to the performance database using SSL. Periodic updating is also used for monitoring the survivability of the measurement nodes. After uploading the measurement results, the measurement data at each node are

removed within a certain period (i.e., a few days) automatically.

– Time synchronization
All measurement nodes synchronize their clocks with the NTP server. To eliminate clock errors, each measurement node locally computes its own time based on the Time Stamp Counter (TSC) register that counts CPU ticks. Correspondence between the values of the TSC register and the actual time is periodically computed by simultaneous measurements for the TSC register value and reference time (i.e., the time of each NTP server). Currently, all measurement nodes are synchronized with one of the CDMA-based NTP servers (stratum 1) that are deployed in the intra-network.

(2) Measurement Management Software
The measurement management software is also implemented for the GNU Linux operating system.

– Measurement execution
The operator chooses the measurement paths by picking up pairs of the sender and the receiver measurement nodes. Measurement parameters of probing packets are defined. Once a new set of measurement parameters has been defined, the measurement management software securely commands the measurement nodes using SSL. During the measurement execution task, the measurement management software is on standby until the end of the measurement or the registration of another measurement task. It ensures that no other measurement is duplicated on the same measurement paths. (Multiple measurements can be performed on the same measurement node.)

– Downloading measurement data
The measurement management software periodically receives the measurement results of 11 metrics, explained above, from each measurement node. The total size of the data at each measurement period is only about 300 Kbytes and thus, simultaneous execution of network probing and downloading measurement data is not expected to interfere with precise performance measurement.
In addition to the 11 metrics, detailed (packet by packet) measurement results can be downloaded at an operator 's request, because measurement data for each probing packet convey much more information that is possibly useful for monitoring network-internal states. The measurement management software specifies the measurement data to be uploaded with its period and path. To avoid congestion due to transmission of detailed data (e.g., the total size of detailed data that contain 1000 packets is about 10 Kbytes) the transmission rate-limiting function is implemented. In addition, a selective download function is also implemented. For example, the detailed measurement data of certain paths within certain measurement periods is

downloaded if each measurement metric exceeds the predefined threshold.

– System configuration
Besides the measurement management tasks described above, maintenance tasks are also performed by the measurement management software. Such tasks have low priority and they take place when no other task is being performed by measurement nodes. An example of a maintenance task is to configure a newly incorporated measurement node on the infrastructure. To do this, a firewall to prevent unauthorized access and basic software configuration are added to the infrastructure. Locating information on the node i.e., which router it connects to is recorded. Other maintenance tasks are related to keep-alive functions such as connectivity checking. If no messages (including measurement result uploading) are received within a predefined period, the measurement nodes are assumed to be down and alarm messages are then sent to the operators.

3.2 OSPF Monitoring Framework

OSPF is a link-state routing protocol, meaning that each router within the OSPF area discovers and builds an entire view of the network topology. Using the topology graph, each router computes a shortest-path tree with itself as the root and applies the results to build its forwarding table according to the cost associated with each link. Hence, a measurement node running OSPF can also build an entire view of the network topology by passively capturing LSAs from a router located in each OSPF area. Note that it makes no difference wherever a monitoring point is located in an OSPF area because all OSPF routers in the same area store the same LSAs.

When OSPF is run on a large-scale network, an OSPF domain is usually divided into multiple areas for scalability. For example, an OSPF domain is often organized into areas containing several dozen routers and a backbone area (area 0). Each area must contain at least one router with an interface in the backbone area. Because LSAs are flooded to all OSPF routers in the same area but are not flooded to other areas, for a multi-area OSPF network, the OSPF monitoring framework needs to distribute at least one OSPF monitoring point to each area, we refer to this as an "OSPF monitor," and capture LSAs. Figure 3 shows the system components of the OSPF monitoring framework. Each OSPF monitor is managed by a central OSPF monitoring management software and LSAs captured by each OSPF monitor are uploaded to the topology database.

3.3 Data Analysis Framework

(1) Route Calculation along Measurement Path
Once a new performance measurement task is executed, the data analysis framework calculates the route along the measurement path (defined as a set of sender

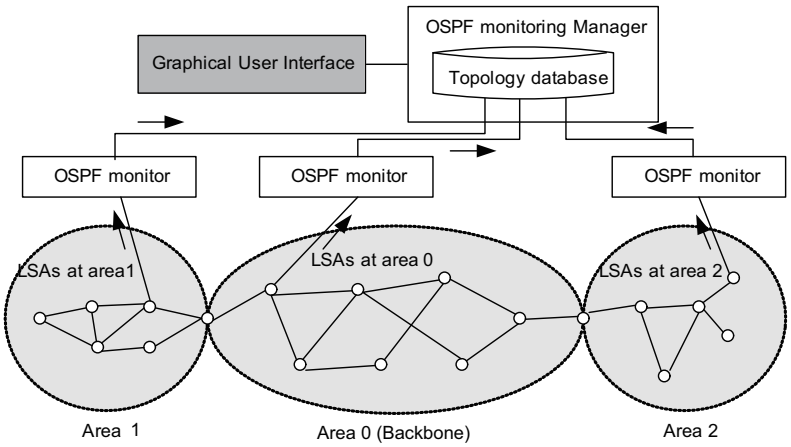

Fig. 3. System components of the OSPF monitoring framework

and receiver measurement nodes). Because the performance measurement framework records the location of each measurement node with the information of the router to which the node connects, the data analysis framework looks up the topology database and finds the shortest route from the router that connects to the sender measurement node to the router that connects to the receiver measurement node according to Dijkstra's algorithm used in OSPF. As mentioned in Section 2, performance data related to the route information itself is often useful for troubleshooting when performance deterioration occurs and it is indispensable to locate performance-degraded segments, as explained below.

(2) Detection of Performance Deterioration

The data analysis framework monitors network packet loss rate and the 95th percentile of delay variation along monitored paths by periodically (e.g., every 1 minute) looking up the performance database. A path is regarded as being degraded within a certain measurement period if either or both of two performance metrics, packet loss rate and the 95th percentile of delay variation, over the time period is larger than a pre-defined threshold (e.g., 0.01 and 10ms, respectively). In this paper, we refer to a path on which performance deterioration is detected as a "deteriorated path." If we detect at least one deteriorated path, the following inference methods are applied.

(3) Locating Performance-degraded Segments

– **Inference method based on Packet Loss Measurements**[17]
If we detect performance deterioration by packet loss, the method classifies all monitored paths' states into three path states, i.e., bad, medium, and good, for the measurement period, using the high and low thresholds of

packet loss rates (e.g., 0.01 and 0.005). By comparing the path statuses, our method infers a candidate set of performance-degraded segments along the deteriorated paths with the following properties: (i) each good path includes none of the segments in the candidate set; (ii) each bad path includes at least one segment in the candidate set; or (iii) the number of segments in the candidate set is the minimum among all possible performance-degraded segment sets satisfying properties (i) and (ii). The method defines a performance-degraded segment as packet loss rate on a segment that is larger than the low threshold (0.005).

– **Inference method based on Clustering the Delay Performance**[18]
If we detect at least one deteriorated path by the 95th percentile of delay variation, clustering of the monitored paths based on the delay variations detected by the probing packets is performed. The method compares the time series of packet delay variation among multiple paths and classifies each path into a cluster by using a hierarchical clustering technique. This technique is designed to effectively identify correlation among delay variations on these paths in conjunction with a network tomographic approach. After clustering, the method can extract all the clusters with at least one deteriorated path. In other words, a set of paths under the following conditions is regarded as a set of degraded paths resulting from a common congestion: the set contains at least one deteriorated path, and each path in the set is subject to similar variations in packet delay in a synchronized manner. The final step involves locating the performance-degraded segments by combining the clustering results with the topology information. During each measurement period, the segments shared by all paths that belong to the same cluster, including some deteriorated paths, are inferred as being performance degraded, and such segments are the cause of those deteriorated paths.
Because the method clusters multiple monitored paths based on time-series of packet delay variation, we need to download detailed (packet-by-packet) measurement data on multiple paths. However, since our concern is which monitored paths are degraded in a synchronized manner, we do not have to compare all monitored paths. Namely, among such paths whose delay variations are very small (completely non-degraded paths), some of them may be eliminated without decreasing the accuracy of the inference results. Actually, in the preliminary analysis, we could reduce the number of paths for which data need to be downloaded, by excluding the paths for which the maximum delay variations in each 60-second period are distinguishably smaller than the candidates of degraded paths.

4 Evaluation and Discussion

In this section, we evaluate the three functionalities of our infrastructure, performance measurement, OSPF monitoring and locating performance-degraded segments. The accuracy of performance measurement is verified by deploying a prototype of our infrastructure on a commercial intra-network. Since it is difficult to actually induce OSPF routing failure and performance deterioration in a

real intra-network, these two functionalities are verified on a local experimental network.

4.1 Measurement Accuracy

Measuring IP performance accurately is the most essential part of our management infrastructure. Accuracy-verifying tests were conducted over the commercial intra-network. The core nodes of the network are connected with multiple 10 Gigabit/Gigabit lines.

We chose 4 paths and made active measurements along them for 2 days. The distance of the targeted paths is in the range from about 1000 km and 1500 km. To verify measurement accuracy, we deployed another hardware-based (more expensive) measurement system "IQ2000 [19]," which actively measures end-to-end IP performance on a targeted path more accurately than a software-based system. Probing packets with an interval of 20 ms are sent on both measurement infrastructures. Table 1 shows the distribution of the difference in the 95th percentile of one-way packet delay on a targeted path within 1 minute measured by our infrastructure and IQ2000. In addition, the distribution of the differences in packet loss rates measured is also indicated. The following results were obtained.

- The average of the difference in the 95th percentile of one-way packet delay was 0.17 ms. In about 99.3% (2783/2804) of 1-minute periods, the difference of the 95th percentile of one-way delay was less than 1 ms.
- In about 99.7% (2795/2804) of 1-minute periods, the difference in packet loss rates was less than 0.004.
- Measurement results along the other paths were roughly the same. The proportion of the periods within which the difference in the 95th percentile of one-way packet delay was less than 1 ms ranged from 99.1% to 99.8%. On the other hand, the proportion of periods within which the difference of packet loss rate was less than 0.004 ranged from 99.3% to 99.8%.

Table 1. Distribution of the difference between the two measurement infrastructures

D_d is the difference of packet delay [ms]	the number of 1-minute periods
$0 \leq \lvert D_d \rvert < 0.25$	2588
$0.25 \leq \lvert D_d \rvert < 0.5$	183
$0.5 \leq \lvert D_d \rvert < 0.75$	6
$0.75 \leq \lvert D_d \rvert < 1$	6
$1 \leq \lvert D_d \rvert$	21

D_l is the difference of loss rate	the number of 1-minute periods
$0 \leq \lvert D_l \rvert < 0.001$	2603
$0.001 \leq \lvert D_l \rvert < 0.002$	131
$0.002 \leq \lvert D_l \rvert < 0.003$	44
$0.003 \leq \lvert D_l \rvert < 0.004$	17
$0.004 \leq \lvert D_l \rvert$	9

4.2 OSPF Monitoring

We established an experimental network consisting of five OSPF running routers (a), (b), (c), (d) and (e), the OSPF monitor and four performance measurement nodes as illustrated in Fig. 4. The OSPF monitor captures LSAs by establishing the adjacency with a router (b) and draws the entire topology of the experimental network. Figure 5 shows the topology drawn by our infrastructure with the graphical user interface. As a result, we could confirm that monitoring LSAs precisely draw the real router-level topology. In addition, we investigated behavior when links/routers are removed from the experimental environment. We monitored the topology via the graphical user interface when one of links "(a)-(b)", "(b)-(c)", "(b)-(d)" and "(d)-(e)", or one of routers (a), (c), (d) and (e) is removed. The results are summarized as follows.

- In all cases, we could confirm that monitoring OSPF precisely recognizes the removal of links/routers and redraws the new topology.
- The interval before our system recognizes a routing failure was in the range from 5 to 45 seconds. The interval mainly depends on when each router detects a failure. In this experiment, the hello-interval, the interval between the sending time of OSPF Hello Packets, and dead-interval, a timer used to time out inactive adjacencies, on each router were 10 and 40 seconds (default values of the router), respectively.

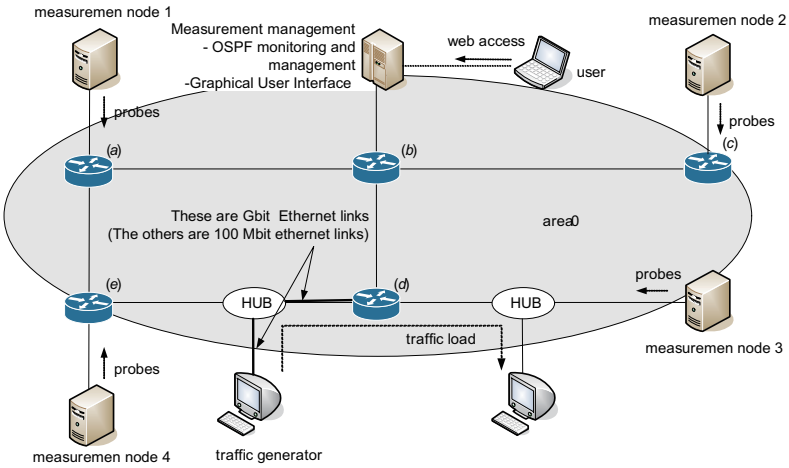

Fig. 4. Experimental local network

Extension for OSPF Equal-Cost Multipath

Multipath routing "Equal-Cost Multipath (ECMP)" is implemented with the

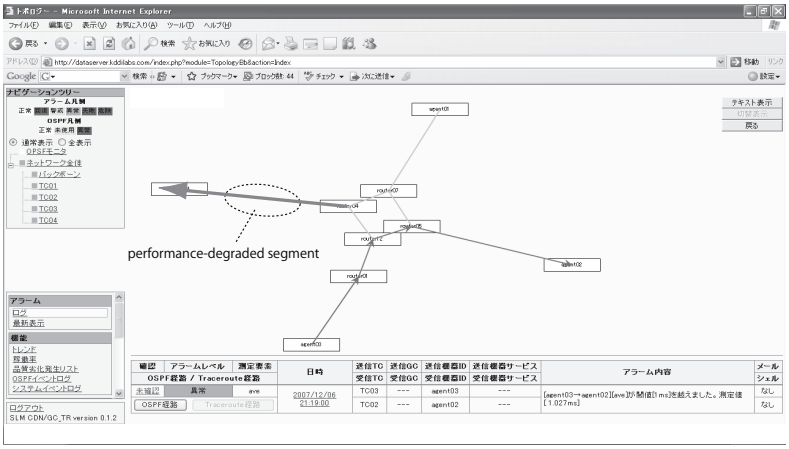

Fig. 5. An example of monitoring OSPF

OSPF. ECMP splits the traffic evenly among equal-cost paths to each destination by distributing the packets among them. Since the OSPF monitoring infrastructure monitors only the LSAs, the current version of our infrastructure cannot ascertain which paths each packet passes through at the ECMP branch points. On the other hand, the hash-based version of ECMP that maps packets from the same flow to the same path is often performed to avoid packet reordering [20, 21]. For this type of ECMP routing, measuring the measurement paths by periodically issuing the traceroute command specifying the same UDP port number as the network probing packets use, in addition to the OSPF monitoring task, might be able to obtain more detailed route information.

4.3 Inference of Performance-degraded segments

To verify the functionality of locating performance-degraded segments, we chose a targeted segment from router (d) to measurement node 3, and caused congestion by loading heavy traffic on the segment. In Fig. 4, the traffic generator sent a number of packet trains, each of which consists of several dozen 1500 byte UDP packets. The interval of each packet train is uniformly distributed from 1 to 10 ms. By continuously monitoring the inference results shown on the graphical user interface as illustrated in Fig. 5, we checked whether the targeted segments are inferred as being degraded or not (the performance-degraded segment is indicated by bold gray arrow). Here, we define the performance-deterioration as the condition under which either or both of the following two conditions is satisfied: (i) Packet loss rate is larger than 0.01; (ii) The 95th percentile of packet delay variation is larger than 10 ms.

The results are summarized as follows.

– During heavy traffic loading, there was a total of 131 1-minute periods during which performance deterioration was detected. (46 and 109 1-minute periods were detected by packet loss rate and delay variation, respectively.)
– In 38 (29%) of the 131 1-minute periods, the data analysis framework failed to infer performance-degraded segments because the inference methods stop when any possibility of miss-inference is recognized to avoid inferring wrong segments. In 86 (92%) of the remaining 93 periods, the targeted segments were inferred as being congested correctly. In all cases, inference was performed in 1 minute or less.

5 Conclusion

In this paper, we proposed an IP performance management infrastructure having the twin frameworks of performance measurement and topology monitoring. By combining above frameworks, the infrastructure locates performance-degraded segments. Through an experiment over a commercial intra-network and an experimental local network, we verified that the proposed infrastructure can manage IP performance, topology and the location of the performance-degraded segments precisely and in real time.

References

1. A.Shaikh and A.Greenberg, "OSPF Monitoring: Architecture, Design and Deployment Experience," In Proc. USENIX Symposium on Network Systems and Design and Implementation (NSDI), 2004.
2. A.Shaikh, M.Goyal, A.Greenberg, R.Rajan and K.Ramakrishnan, "An OSPF Topology Server: Design and Evaluation." IEEE J.Selected Areas in Communications, vol.20, no.4, 2002.
3. D.Watson, C.Labovitz and F.Jahanian, "Experiences with Monitoring OSPF on a Regional Service Provider Network," In Proc. IEEE International Conference on Distributed Computing Systems (ICDCS), 2003.
4. K.C. Claffy, Tracie E. Monk, and Daniel McRobb, "Internet tomography." Nature, Web Matter. January 1999.
5. L. Peterson, T. Anderson, D. Culler, and T. Roscoe, "A blueprint for introducing disruptive technology into the Internet," In Proc. ACM Workshop on Hot Topics in Networks (HotNets), Princeton, NJ, October 2002, pp. 59-64.
6. D.G. Andersen, H. Balakrishnan, M.F. Kaashoek, and R. Morris, "Resilient overlay networks," In Proc. ACM Symposium on Operating Systems Principles (SOSP), Banff, Alberta, Canada, October 2001, pp. 131-145.
7. C.R. Simpson, Jr. and G.F. Riley, "NETI@home: A distributed approach to collecting end-to-end network performance measurements," In Proc. Passive & Active Measurement (PAM), Antibes Juan-les-Pins, France, April 2004.
8. Y. Shavitt and E. Shir, "DIMES: Let the internet measure itself." SIGCOMM Computer Communication Review, vol. 35, no. 5, pp. 71-74, 2005.
9. S. Kalidindi and M.J. Zekauskas, "Surveyor: An infrastructure for Internet performance measurements," In Proc of INET'99, June 1999.
10. "Active Measurement Project," http://amp.nlanr.net/.

11. V. Paxson, A. Adams, and M. Mathis, "Experiences with NIMI," In Proc. Passive & Active Measurement (PAM), April 2000.
12. R. Caceres, N. Duffield, J. Horowitz, and D. Towsley, "Multicast-based inference of network-internal loss characteristcs." IEEE Trans. Info. Theory, 45(7):2462–2480, 1999.
13. N. Duffield, F. L. Presti, V. Paxson, and D. Towsley, "Inferring link loss using striped unicast probes," In Proc. IEEE infocom, Anchorage, April 2001.
14. M. Tsuru, T. Takine, and Y. Oie, "Inferring link characteristics from end-to-end path measurements," In Proc. IEEE ICC, pages 1534–1538, Helsinki, June 2001.
15. A. Bestavros, J. Byers, and K. Harfoush, "Inference and labeling of metric-induced network topologies," In Proc. IEEE infocom, New York, June 2002.
16. V. N. Padmanabhan, L. Qiu, and H. Wang, "Server-based inference of internet link lossiness," In Proc. IEEE infocom, San Francisco, April 2003.
17. A. Tachibana, S. Ano, T. Hasegawa, M. Tsuru, and Y. Oie, "Locating Performance-degraded segments over the Internet Based on Multiple End-to-End Path Measurements." IEICE Transactions on Communications Internet Technology VI, April 2006.
18. A. Tachibana, S. Ano, T. Hasegawa, M. Tsuru, and Y. Oie, " Locating Performance-degraded segments on the Internet by Clustering the Delay Performance of Multiple Paths," In Proc. IEEE ICC, Glasgow, June 2007.
19. IQ2000 QoS Monitoring system, Yokogawa Electric Corporation, http://www.yokogawa.com/tm/data/tm-data.htm
20. Cisco express forwarding(cef). Cisco white paper, Cisco Systems., July 2002.
21. Junos 6.3 internet software routing protocols configuration guide. www.juniper.net/techppubs/software/junos/junos63/swconfig63-routing/html/.