PAPER Special Section on Parallel and Distributed Computing and Networking

Traffic Anomaly Analysis and Characteristics on a Virtualized Network Testbed

Chunghan LEE[†*a)], **Hirotake ABE**[††b)], **Toshio HIROTSU**[†††c)], *Nonmembers*, *and* **Kyoji UMEMURA**[††††d)], *Member*

SUMMARY Network testbeds have been used for network measurement and experiments. In such testbeds, resources, such as CPU, memory, and I/O interfaces, are shared and virtualized to maximize node utility for many users. A few studies have investigated the impact of virtualization on precise network measurement and understood Internet traffic characteristics on virtualized testbeds. Although scheduling latency and heavy loads are reportedly affected in precise network measurement, no clear conditions or criteria have been established. Moreover, empirical-statistical criteria and methods that pick out anomalous cases for precise network experiments are required on userland because virtualization technology used in the provided testbeds is hardly replaceable. In this paper, we show that 'oversize packet spacing', which can be caused by CPU scheduling latency, is a major cause of throughput instability on a virtualized network testbed even when no significant changes occur in well-known network metrics. These are unusual anomalies on virtualized network environment. Empirical-statistical analysis results accord with results at previous work. If network throughput is decreased by the anomalies, we should carefully review measurement results. Our empirical approach enables anomalous cases to be identified. We present CPU availability as an important criterion for estimating the anomalies.

key words: *network measurement, virtualization, PlanetLab*

1. Introduction

Network testbeds, e.g., Emulab [1], StarBED [2], and PlanetLab [3], [4], have been used for network researches, distributed system researches, and network experiments. In such testbeds, resources, such as CPU, memory, and I/O interfaces, are shared and virtualized to maximize node utility for many users. PlanetLab, which is a virtualized network testbed over the Internet, has been used to investigate the validity of measurement tools [5] and prediction methods [6]. We have found that this kind of prediction needs precise measurements to obtain the learning samples. As of January 2011, it has grown to 1,100 machines spanning more than 500 sites and 40 countries. In PlanetLab, a node based on the Linux-VServer shares the resources. A platform called a *sliver* is provided as a virtualized environment to users and multiple slivers can be run simultaneously at each node. A set of these slivers participating in the same activity at different nodes is called a *slice*. Thus, PlanetLab consists of virtualized nodes on the Internet.

A few studies [7], [8] have investigated the impact of virtualization on precise network measurement and have understood Internet traffic characteristics on the virtualized testbed. Although CPU scheduling latency and heavy loads are reportedly affected in precise network measurement, no clear conditions or criteria have been established. In virtualized testbeds, real resource state on a physical node is hidden and it is hard to track user behaviour on the same node because under layer is black-box. Moreover, virtualization technology used in the provided testbeds is hardly replaceable. Empirical-statistical criteria and methods that pick out anomalous cases for precise network experiments are required on userland.

The aims of our study are to investigate the impact of virtualization for precise network experiments, to clarify Internet traffic characteristics on a virtualized testbed, and to determine clear conditions and criteria for estimating anomalous cases. These goals are required to establish a throughput prediction method for network virtualization. To achieve these goals, we are using PlanetLab as a virtualized network testbed. We measure network throughput to clarify Internet traffic characteristics on the virtualized testbed and to find useful parameters for predicting network throughput. In the throughput measurement, we use a pair of different-sized connections called as 'connection pair'. In order to identify conditions and criteria for estimating the anomalous cases, we monitor resource states with a simple monitoring program that consumes CPU cycles during the generation of connection pair.

In this paper, we present the empirical-statistical analysis of throughput measurement and resource monitoring to estimate anomalous cases in precise network experiments on the virtualized testbed. The main contributions of this work are:

- We show that 'oversize packet spacing', which can be caused by CPU scheduling latency, is a major cause of throughput instability on the virtualized testbed even when no significant changes occur in well-known network metrics.
- We analyze measurement results to discard anomalous

Manuscript received January 5, 2011.
Manuscript revised June 10, 2011.

[†]The author is with the Dept. of Electronic and Information Eng., Toyohashi University of Tech., Toyohashi-shi, 441–8530 Japan.
[††]The author is with Cybermedia Center, Osaka University, Ibaraki-shi, 567–0047 Japan.
[†††]The author is with the Faculty of Computer and Information Sciences, Hosei University, Koganei-shi, 184–0002 Japan.
[††††]The author is with the Dept. of Computer Science and Eng., Toyohashi University of Tech., Toyohashi-shi, 441–8530 Japan.
a) E-mail: lch@ss.cs.tut.ac.jp
b) E-mail: habe@cmc.osaka-u.ac.jp
c) E-mail: hirotsu@hosei.ac.jp
d) E-mail: umemura@tut.jp

DOI: 10.1587/transinf.E94.D.2353

cases empirically. Empirical-statistical analysis results accord with the results shown by Peterson et al. [7]. The oversize packet spacings are unusual anomalies on virtualized network environment.

- Our empirical approach that observes criteria provided by system during the throughput measurement and analyzes the criteria statistically enables anomalous cases to be identified on the virtualized testbed. Moreover, we find that CPU availability is an important criterion for estimating the anomalies.

The rest of this paper is organized as follows. First, we describe related work in Sect. 2. We present necessary for estimating anomalous cases on userland in Sect. 3. Our measurement methodology is explained in Sect. 4. We present the throughput measurement and discuss the empirical-statistical analysis results of network metrics in Sect. 5. We show the empirical-statistical analysis results of resource monitoring in Sect. 6. Finally, we conclude the paper with a summary of the main points in Sect. 7.

2. Related Work

2.1 Fluctuation in Packet Spacing on Virtualized Testbed

Peterson et al. [7] deployed a packet forwarding overlay between Seattle and Washington, D.C. on the virtualized testbed and used ping packets to compare the RTT between the Seattle and D.C. nodes for the network and overlay. The RTT (Round Trip Time) of network was constant while that of the overlay varied widely. The cause of the fluctuation in RTT was CPU scheduling latency. Although the scheduling latency at a node will be a serious problem for network applications, no consideration has been given to the relationship between packet spacing fluctuation and scheduling latency.

Spring et al. [8] showed that the load prevents accurate latency measurement and precise spacing for packet trains. They ran traceroute and tcpdump in parallel to acquire timestamps between the application and kernel levels and showed the differences between application and kernel-captured timestamps when sending probes and receiving responses. Moreover, they transmitted packet trains to determine how the CPU load impaired precisely-spaced packets. However, they showed no clear conditions for the type of load. In their analyses, they did not discuss how to estimate anomalous cases in network experiments.

2.2 Monitoring Systems

CoMon [9], [10] is a centralized resource monitoring system for the virtualized testbed. It provides views of the virtualized testbed, such as node-centric and slice-centric information. Moreover, it has been used for selecting nodes and identifying problems on the virtualized testbed. Because it gathers data every five minutes, the data granularity is limited and the data type makes it hard to estimate fluctuation in packet spacing.

Clue [11] is an anomaly detection system for the virtualized testbed. However, this system focused on detecting anomalous behavior for the virtualized testbed and it used data on CoMon only.

Slicestat [12] provides slice-level resource consumption information, such as CPU and memory utilization, network I/O, number of processes, and so on, at each node on the virtualized testbed. It does not provide node-level information, such as SSH failing and shutdown. In these monitoring systems, however, the authors did not discuss any anomalous cases occurring in their network experiments.

3. Necessary for Estimating Anomalous Cases on Userland

Virtualized network testbeds become popular to make network experiments with ease. In system construction on non-virtualized environment, we can re-implement under layer on the system. However, the situation on virtualized testbeds is different from the non-virtualized environment. Thus, virtualization technology used in the provided testbeds is hardly replaceable. Moreover, the under layer on the virtualized testbeds is normally black-box. Real resource state on a physical node is hidden and it is hard to track user behaviour on the same node over virtualized testbed. Empirical-statistical criteria and methods that pick out anomalous cases for precise network experiments are required on userland.

4. Measurement Methodology

4.1 Node Selection

We empirically select four node pairs (eight nodes), which we refer to as nodes (α, β), (γ, δ), (κ, λ), and (μ, ν). These nodes are contributed by universities or institutions across North America and Europe participating to PlanetLab, and composed of two or four independent CPU cores and physical memory of approximately 3 GB (gigabytes). The virtualized network testbed does not support network virtualization for virtualized networks, thus the resource is shared among slices. It would be possible to use it at each site, but we did not consider it in the experiments. We measure network throughput using Iperf and RTT using ping; their basic characteristics are given in Table 1. We observed throughput instability and abrupt RTT fluctuation in node pairs for two pairs (κ, λ) and (μ, ν). To show the abrupt RTT fluctuation, we introduce RTT using ping at pair (κ, λ) at Fig. 1. We find an abrupt value of RTT (0.3990 s), and it is larger than the others.

4.2 Throughput Measurement with Resource Monitoring

Our objective in measuring throughput is to identify Internet traffic characteristics for network throughput prediction on network virtualization areas. A prediction method has

Table 1 Node characteristics on virtualized network testbed.

Node name	CPU cores	Memory [GB]	Node pair (arrow denotes transfer direction)	Mean RTT [s]
α	2	2.96	$\alpha \leftarrow \beta$	0.0428
β	4	3.46		
γ	4	3.42	$\gamma \leftarrow \delta$	0.0830
δ	4	3.21		
κ	4	3.42	$\kappa \leftarrow \lambda$	0.0200
λ	2	3.47		
μ	2	3.45	$\mu \leftarrow \nu$	0.0900
ν	2	2.97		

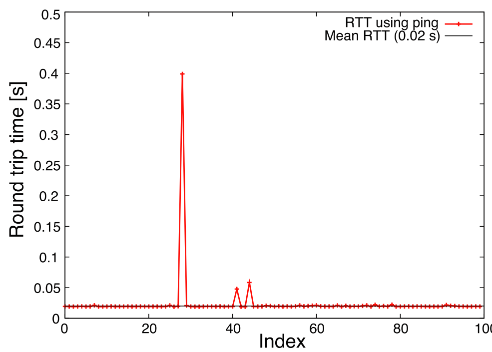


Fig. 1 RTT using ping at pair (κ, λ).

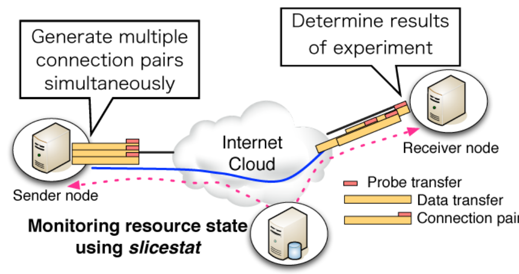


Fig. 2 Connection pair measurement method.

Table 2 Probe and data size combinations for connection pairs.

Index	Probe size [KB]	Data size [MB]	Number of connection pairs
1	16	8	3
2	16	16	3
3	32	16	3
4	64	16	3
5	64	32	3

been previously proposed using a pair of different-sized connections. This method, which we call 'connection pair', uses a small sized probe transfer to predict the throughput of a large sized data transfer. Wolski et al. [13], [14] used different-sized pairs of connections and empirically established the basic probe size as 64 KB (kilobytes) for Network Weather Service (NWS). However, they did not consider the appropriate size for probe transfer on the current Internet. We used multiple TCP connections to press router's queue at end-to-end path and to encounter congested network state. In related work [15], when the number of connections was over 8 on non-virtualized testbed, the throughput was saturated. In the virtualized testbed, there are various kinds of network traffic of the other slices on the node. We considered the above situation, and used 6 connections (3 connection pairs) simultaneously. We generated the various sizes for the connection pair every 5 minutes at all the pairs, and monitored the resource state per slice on the node with slicestat every 1 minute during throughput measurement. If the measured size is smaller than expected or if the transfer time is more than 5 minutes, we judge at the receiver that the experiment has failed. TCP window sizes for probe and data transfers are respectively reduced to 16 and 64 KB. These window sizes for the connection pair have been used in NWS. The measurement methodology is described in Fig. 2. The sizes for the connection pair are shown in Table 2. These are a part of sizes used in our previous work [16].

Although CPU scheduling latency [7] and heavy loads [8] are likely to affect the network measurement, they did not investigate the impact of virtualization over many sites and no clear conditions or criteria had been established to estimate unstable conditions. Moreover, it is hard to directly observe CPU scheduling latency on the virtualized network testbed. Our approach is to implement a simple CPU monitoring program that gets the current time every second in order to investigate CPU availability. It consists of a loop for timestamp acquisition. In each iteration, the monitoring program calls get-timeofday() and the timestamp is saved in a pre-allocated memory. If the resource state is stable, the constant spaced timestamps are stored and one CPU core will be allocated to the monitoring program fully. It is ran during throughput measurement.

5. Measurement Results and Packet-Level Analysis

In this section, we describe the throughput measurement results and packet-level analysis. Our objective in packet-level analysis is to find the cause of the throughput instability and to clarify the Internet traffic characteristics. We analyze well-known network metrics, such as RTT and packet loss rate, for the connection pair where the probe size was 64 KB and the data size was 32 MB in all the node pairs. Next, we investigate packet spacing at the sender and advertised window. Finally, we introduce anomalous cases to understand traffic characteristics on the virtualized testbed.

5.1 Throughput Measurement Results

5.1.1 Theoretical Achievable Throughput

Theoretical achievable throughput is defined as

$$Throughput = \frac{W \cdot p}{P \cdot R} \tag{1}$$

where W is advertised window size, p is packet size without headers, P is packet size with headers, and R is round

Table 3 Ideal, minimum, mean, and maximum throughput.

Node pair	Ideal [KBps]	Min [KBps]	Mean [KBps]	Max [KBps]
α - β	1475.1	893.4	1073.7	1236.5
γ - δ	760.6	680.4	730.6	739.2
κ - λ	3156.7	251.9	288.6	413.1
μ - ν	701.5	128.7	241.9	363.3

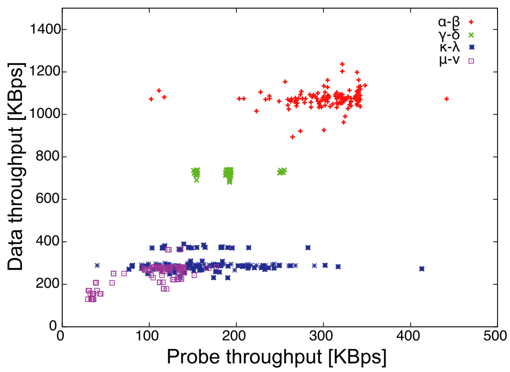


Fig. 3 Throughput measurement results (probe: 64 KB, data: 32 MB).

Table 4 Mean RTT of connection pair.

Node pair	Mean RTT Probe [s]	Mean RTT Data [s]	Mean RTT Ping [s]
α - β	0.0520	0.0556	0.0428
γ - δ	0.0854	0.0829	0.0830
κ - λ	0.0224	0.0225	0.0200
μ - ν	0.0974	0.0974	0.0900

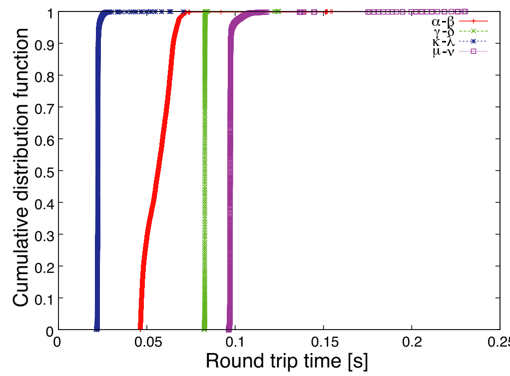


Fig. 4 CDF of RTT of data transfer at all node pairs (32 MB).

Table 5 Mean packet loss rate of connection pair.

Node pair	Mean loss rate [%] (Probe)	Mean loss rate [%] (Data)
α - β	0.032	0.049
γ - δ	0.216	0.015
κ - λ	0	0.001
μ - ν	0	0.001

trip time. In the theoretical achievable throughput, we assume that there is no packet loss and advertised window is saturated to the maximum size. For example, when the advertised window is 64 KB, RTT is 0.02 s, packet size without headers is 1368 Bytes, and packet size with headers is 1420 Bytes, the ideal case using the theory is 3156.7 KBps. We can compare differences in the ideal and the measurement results.

5.1.2 The Statistics of Measurement Results

We gathered a dataset of approximately 2,000 connection pairs from all the node pairs everyday. Table 3 presents the ideal case using the theoretical achievable throughput and the statistics when the probe size was 64 KB and the data size was 32 MB (megabytes). The parameters, such as packet size, header size and so on, of the ideal are equal to the measurement results. The mean throughput at pairs (α, β), (κ, λ), and (μ, ν) is different from the ideal case. Moreover, the maximum throughput at pairs (κ, λ) and (μ, ν) is not close to the ideal case. It is very different in comparison with the maximum value at pairs (α, β) and (γ, δ). Moreover, we find a significant difference in the statistics at pairs (κ, λ) and (μ, ν). Figure 3 shows throughput measurement results of the above size. Also, the decreases in network throughput are observed on two pairs (κ, λ) and (μ, ν).

5.2 RTT

The RTT normally fluctuates and network throughput decreases as a result of network congestion on the end-to-end path. Table 4 shows the mean RTT for all the node pairs. We find increases in the mean RTT at pair (α, β) in comparison with the mean RTT using ping. However, the range of fluctuation of the mean RTT at the others is smaller than the case at pair (α, β). To investigate the fluctuations in RTT, we present the cumulative distribution function (CDF) of RTT at Fig. 4. The values of throughput at all the pairs (α, β), (γ, δ), (κ, λ), and (μ, ν) are 1085.8 KBps, 738.3 KBps, 399.0 KBps, and 363.3 KBps respectively. In the CDF at pair (α, β), RTT is gradually increased, and it is a cause of fluctuation in throughput. Although the approximate 90% of RTT is close to the mean RTT using ping at two pairs (κ, λ) and (μ, ν), the decreases in network throughput are occurred. Thus, the fluctuations in RTT are not enough to prove the major cause of throughput instability at two pairs (κ, λ) and (μ, ν).

5.3 Packet Loss Rate

The packet loss rate is normally an important metric for network throughput. If it is high, the network throughput is decreased. The mean loss rates at all the pairs are shown in Table 5; they are smaller than 1%. To summarize, both packet loss and the fluctuations in RTT were the major cause of throughput fluctuation at pair (α, β). Conversely, these metrics could not clarify the cause of the throughput instability at two pairs (κ, λ) and (μ, ν). Thus, the cause of the throughput instability at two pairs (κ, λ) and (μ, ν) was not any network effect, such as network congestion, on the virtualized testbed. Moreover, we find that there is no packet loss though the values of RTT are increased. In it, the mean

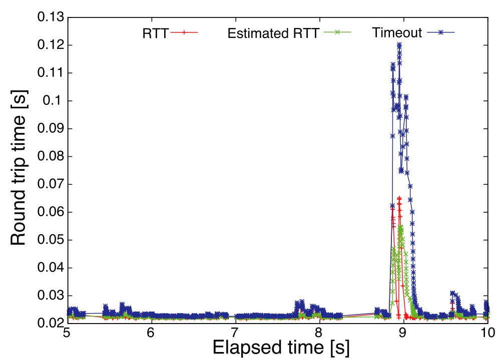


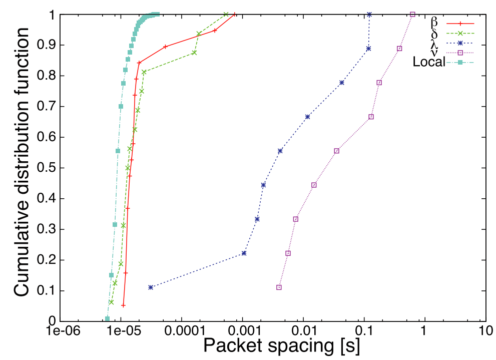
Fig. 5 RTT, Estimated RTT, and Timeout of data transfer at pair (κ, λ).

RTT is 0.0231 s, the mean packet spacing is 0.013366 s, and throughput is 283.5 KBps. To identify the validity of the above case, we investigate TCP's timeout interval [17] of the case. We present RTT, estimated RTT, and the timeout interval of the above case at Fig. 5. Although the values of RTT are increased, the timeout intervals are larger than the values of RTT, and there is no packet loss.

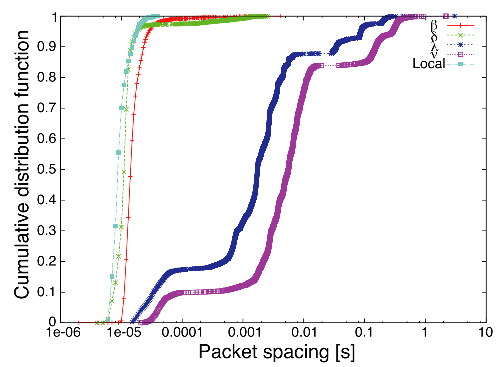
5.4 Cause of Network Throughput Drop

We observed decreases in network throughput for particular pairs. When many connections exceed the acceptable number of connections concurrently concentrated on the receiver node, the network throughput decreases. Rak et al. [18] observed the number of pending requests at a server by increasing the number of requests to evaluate Web Service resilience. However, our case differs from the above case because resources, such as memory and network bandwidth, at each node are limited by a watchdog daemon. This daemon forcibly reduces a user's memory rate when the memory rate suddenly exceeds the threshold amount.

A 'packet spacing' is an idle period between the reception of a packet and the sending of the next packet. The packet spacing is a very short period over non-virtualized environments. To investigate packet spacing over a non-virtualized environment, we considered that a local environment consists of a router and two native nodes. Each native node has an Intel Pentium 4 processor with a 1 GB RAM and a 100 Mbps network card and runs on Ubuntu 9.04. The router was connected between these nodes via a 100 Mbps link. There are no heavy loads at each node and no network traffic between the nodes. We generated the sizes for connection pair in Table 2. When data size was 32 MB, the minimum, mean, maximum values of packet spacing are 0.000006 s, 0.000010 s, 0.000040 s respectively. In the local, there is no abrupt fluctuation in the packet spacing and these values do not affect the throughput measurement. Next, we show the packet spacing on the virtualized testbed. Figure 6 shows the CDF of packet spacing at the local and that for connection pair where the probe was 64 KB and the data was 32 MB at all the pairs. The approximate 80% of packet spacings for the probe transfer and the approximate 90% for the data transfer at nodes β and δ are similar to



(a) Probe transfer (64 KB)



(b) Data transfer (32 MB)

Fig. 6 CDF of packet spacing for connection pair at all pairs.

the packet spacing at the local. In the case at node β, the throughput of the data transfer is 1093.9 KBps, the mean RTT is 0.0559 s, and the mean packet spacing is 0.000021 s. The CDF for connection pair at node β is similar to the CDF of the local. Although a portion of packet spacings at node β are increased, these cannot be the major cause of fluctuations in throughput. The case at node δ is similar to the above case. However, the packet spacings at nodes λ and ν are larger than the other nodes. We introduce the CDF of packet spacing at node λ. In this case, the network throughput is 288.1 KBps, mean RTT is 0.0223 s, and mean packet spacing is 0.014064 s. Although the mean RTT is similar to that using ping, a portion of the packet spacings is larger than the packet transmission period, and the CDF of packet spacing is very different from the case at the local and nodes β and δ. Moreover, we show the mean packet spacing for connection pair where the probe was 64 KB and the data was 32 MB at Fig. 7. The mean packet spacing at nodes λ and ν is larger than the case at nodes β and δ, and these are very different from the case at the local.

The oversize packet spacings were a major cause of throughput instability even when there were no abrupt changes in the well-known network metrics at pairs (κ, λ) and (μ, ν). It accords with results at the previous work [7]. These oversize packet spacings are unusual anomalies. When the packet spacing was larger than RTT, packets at the sender node were not sent consecutively. These rarely happen in non-virtualized network environment while it is easy for anomalies to be occurred in virtualized network environ-

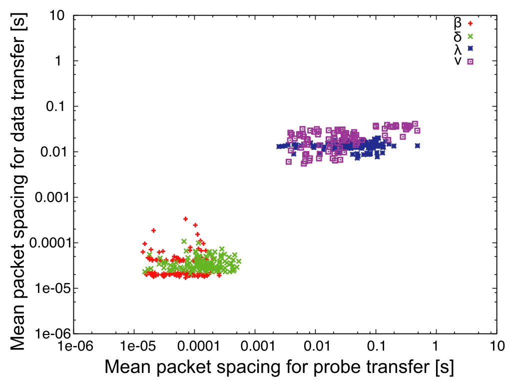


Fig. 7 Mean packet spacing for connection pair at all senders (probe: 64 KB, data: 32 MB).

ment. The problem described above is more severe on the virtualized environments than the non-virtualized environments because resources are virtualized and shared among virtual machines, and it would be hard to monitor the other virtual machines on the node.

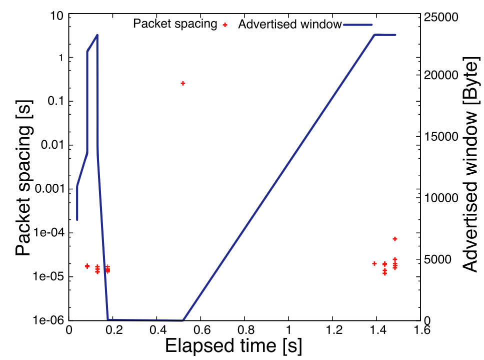
5.5 Advertised Window

TCP provides flow control for the receiver to control transmission speed, so that the receiver is not overwhelmed with data from the sender. The advertised window is used to give the sender an idea of how much free buffer space is available at the receiver. If the advertised window at the receiver is zero, the sender does not send data to the receiver after ACK packet; thus, the packet spacing is increased at the sender. We show two cases at Fig. 8; one case is that the packet spacings are increased and the advertised window is decreased due to flow control while the other case is that the packet spacings are increased and the advertised window is saturated to up maximum size. In it, packet spacings do not relate to flow control.

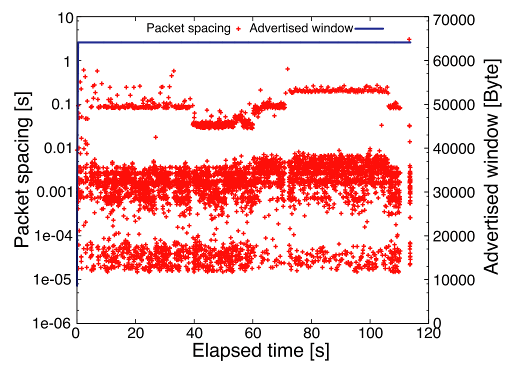
5.6 Anomalous Case

To summarize, the result at pair (α, β) experienced the fluctuations in network throughput caused by the packet loss and the fluctuations in RTT. This case is normally predictable. But, the case at two pairs (κ, λ) and (μ, ν) differed from the above case. In this case, the network throughput drop was occurred by the anomalies with stable network state.

Major distinction of anomalous cases is instability of throughput despite of stable network state, which can be observed through RTT, packet loss rate, and so on. The condition for the judgment of the anomalous case is the impact of packet spacing. We introduce the case at node λ as an example of the anomalous case. We show the CDF of RTT and packet spacing at Fig. 9. In this case, the throughput is 288.6 KBps, the loss rate is zero, the advertised window is saturated to the maximum size (64 KB), and minimum, mean, and maximum RTT are 0.0213 s, 0.0222 s, and 0.0274 s. The statistics of RTT are close to the mean

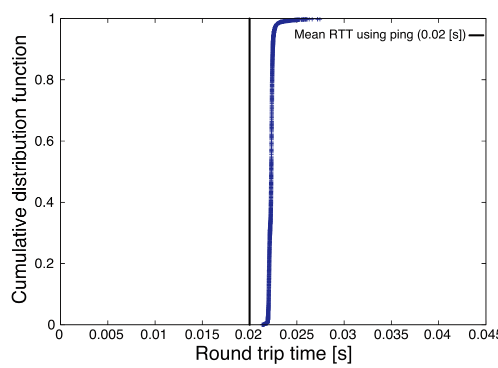


(a) Packet spacing and advertised window of probe transfer at β (64 KB)

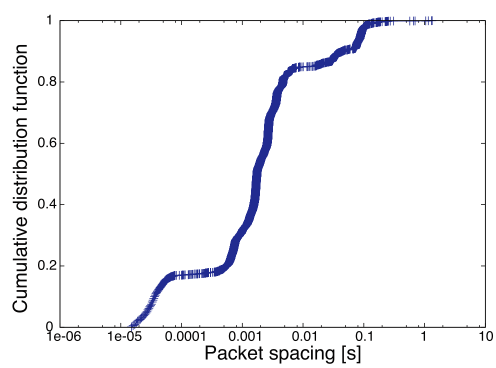


(b) Packet spacing and advertised window of data transfer at λ (32 MB)

Fig. 8 Packet spacing and advertised window.



(a) CDF of RTT



(b) CDF of packet spacing

Fig. 9 CDF of RTT and packet spacing at λ (32 MB).

RTT (0.02 s) using ping. However, minimum, mean, and maximum packet spacing are 0.000015 s, 0.013618 s, and 1.332624 s respectively. The CDF of packet spacing at node λ is different from the CDF at nodes β and δ. To summarize, the most of values of RTT are stable, and the throughput instability are occurred by the anomalies. This case is the anomalous case. If network throughput is decreased by the anomalies, we should carefully review measurement results. We show the cases at pairs (κ, λ) and (μ, ν) as the anomalous cases at Appendix, and the accurate definition is our future work.

6. Resource Monitoring Results

In the previous section, we observed decreases in network throughput at particular pairs. The major causes of the throughput decrease were anomalies. Previous works [7], [8] did not investigate the anomalies at many sites. Moreover, no clear conditions or criteria had been established to estimate the effect of the anomalies. During the generation of the connection pair, we analyzed resource monitoring results to investigate the anomaly condition and to establish efficient criteria for the anomaly estimation.

6.1 CPU Utilization

In a multi-core processor, if there are two or four CPU cores on the processor, maximum CPU utilization can be showed to 200% or 400%. However, these can give us confusion. We divided these values into the number of cores, and normalized these values as 100%. In order to investigate the anomaly condition, we observed CPU utilization at the sender node using slicestat during the connection pair generation. The CPU utilization at all the sender nodes is shown in Fig. 10. The CPU utilization at node δ was lower than the other senders. In this case, there were no anomalies because of very low CPU utilization.

Although there was a difference in the mean packet spacings, the CPU utilization did not reflect the mean packet spacings without node δ. In this case, it will be inappropriate for estimating anomalies on the virtualized testbed. We show the statistics of CPU utilization at all the senders at Fig. 11. In the statistics, the case at node β does not differ from the case at nodes λ and ν. The statistics are not enough to estimate the anomalies.

6.2 CPU Availability

While we ran the monitoring program that consumes CPU cycles at the node, we observed CPU availability at the node using the monitoring program. It is different from the CPU utilization. It is available CPUs to be allocated to users while the CPU utilization is used to users at the moment. If there are two CPU cores at the node and the CPUs are idle, one will be fully allocated to the monitoring program. Thus, it uses one CPU core and the CPU utilization can be increased to 50%. Conversely, it will be hard to allocate the CPU to the monitoring program if the CPUs are busy. Figure 12 shows the CPU availability at all the sender nodes. There is a significant change in the CPU availability. Although there were two CPU cores at nodes λ and ν, it was not increased to approximately 50%. These nodes had the anomalies. Conversely, there are four CPU cores at node δ. It is keeping up the approximately 25% and the small packet spacing. The CPU availability is an important criterion for estimating anomalies on the virtualized network testbed. When the CPU utilization is high and the CPU availability is low, the anomalies are occurred.

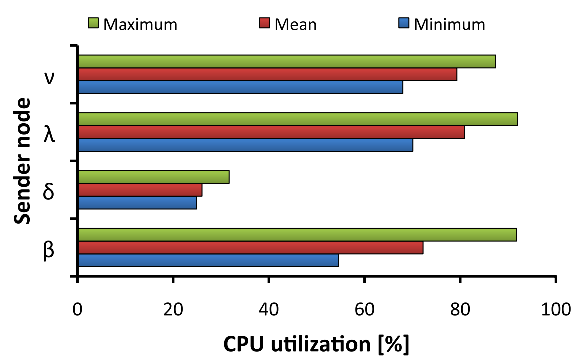


Fig. 11 Minimum, mean, and maximum CPU utilization at all sender nodes.

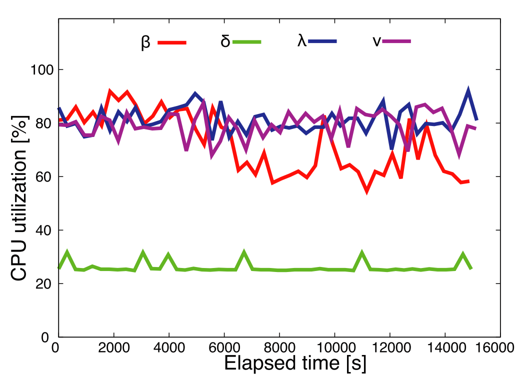


Fig. 10 CPU utilization at all sender nodes.

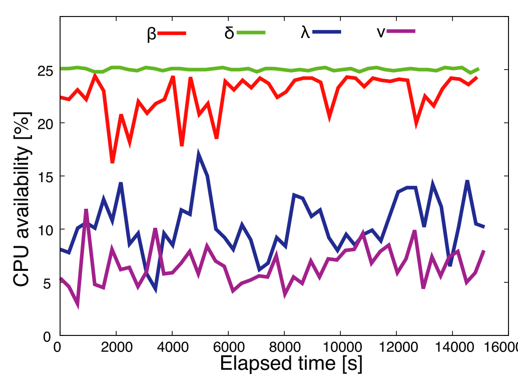


Fig. 12 CPU availability at all sender nodes.

7. Conclusion

This paper describes how we measured and analyzed throughput measurement results to clarify Internet traffic characteristics on the virtualized testbed and how we monitored resources to estimate anomalous cases for precise network experiments.

In measuring throughput, we showed that oversize packet spacing, which can be caused by CPU scheduling latency, is a major cause of throughput instability on the virtualized testbed even when no significant changes occur in the well-known network metrics. Some of the packet spacings were larger than the packet transmission period. These oversize packet spacings were unusual anomalies on virtualized network environment. Empirical-statistical analysis results accord with results at the previous work [7]. We should carefully review measurement results obtained under such the anomalous cases. In monitoring resources, we showed that the anomalies are occurred when the CPU utilization is high and the CPU availability is low. Our empirical approach that observes criteria provided by system during the throughput measurement and analyzes the criteria statistically enables anomalous cases to be identified on the virtualized testbed. We found that the CPU availability is an important criterion for estimating the anomalies.

In future work, we will define the anomalous case accurately, devise a method of compensating for the impact of anomalies, and find more efficient criteria than the CPU availability for anomaly estimation.

Acknowledgments

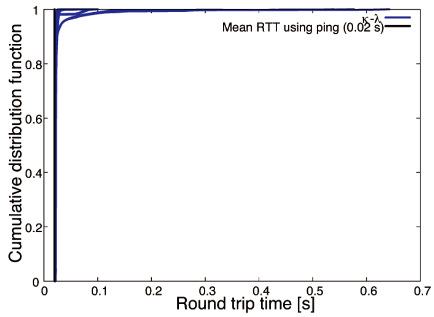
We would like to express our gratitude to the reviewers who reviewed our paper and provided many comments that helped to improve our paper. This work was supported in part by JSPS KAKENHI (22500072 and 22700029) and Global COE Program "Frontiers of Intelligent Sensing" from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

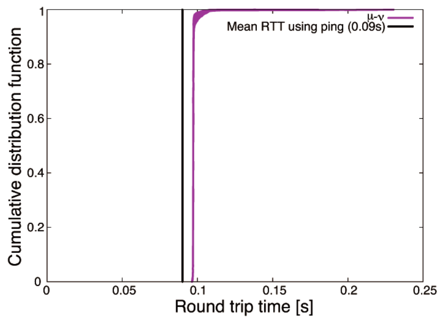
[1] Emulab, http://www.netbed.org/
[2] StarBED, http://www.starbed.org/
[3] PlanetLab, https://www.planet-lab.org/
[4] A. Bavier, M. Bowman, B. Chun, D. Culler, S. Karlin, S. Muir, L. Peterson, T. Roscoe, T. Spalink, and M. Wawrzoniak, "Operating system support for planetary-scale network services," NSDI'04: Proc. 1st Conference on Symposium on Networked Systems Design and Implementation, pp.19–19, Berkeley, CA, USA, 2004.
[5] J. Sommers and P. Barford, "An active measurement system for shared environments," IMC '07: Proc. 7th ACM SIGCOMM Conference on Internet Measurement, pp.303–314, New York, NY, USA, 2007.
[6] D. Lu, Y. Qiao, P.A. Dinda, and F.E. Bustamante, "Characterizing and predicting tcp throughput on the wide area network," ICDCS '05: Proc. 25th IEEE International Conference on Distributed Computing Systems, pp.414–424, Washington, DC, USA, 2005.
[7] L. Peterson, A. Bavier, M.E. Fiuczynski, and S. Muir, "Experiences building planetlab," OSDI '06: Proc. 7th Symposium on Operating Systems Design and Implementation, pp.351–366, Berkeley, CA, USA, 2006.
[8] N. Spring, L. Peterson, A. Bavier, and V. Pai, "Using planetlab for network research: myths, realities, and best practices," SIGOPS Oper. Syst. Rev., vol.40, no.1, pp.17–24, 2006.
[9] CoMon, http://comon.cs.princeton.edu/.
[10] K. Park and V.S. Pai, "Comon: a mostly-scalable monitoring system for planetlab," SIGOPS Oper. Syst. Rev., vol.40, no.1, pp.65–74, 2006.
[11] A. Chandra, R. Prinja, S. Jain, and Z. Zhang, "Co-designing the failure analysis and monitoring of large-scale systems," SIGMETRICS Perform. Eval. Rev., vol.36, no.2, pp.10–15, 2008.
[12] Slicestat, http://codeen.cs.princeton.edu/slicestat/.
[13] M. Swany and R. Wolski, "Multivariate resource performance forecasting in the network weather service," Supercomputing '02: Proc. 2002 ACM/IEEE conference on Supercomputing, pp.1–10, Los Alamitos, CA, USA, 2002.
[14] R. Wolski, N. Spring, and J. Hayes, "The network weather service: A distributed resource performance forecasting service for metacomputing," J. Future Generation Computing Systems, vol.15, pp.757–768, 1999.
[15] T.J. Hacker, B.D. Athey, and B. Noble, "The end-to-end performance effects of parallel tcp sockets on a lossy wide-area network," IPDPS '02: Proc. 16th International Parallel and Distributed Processing Symposium, p.314, Washington, DC, USA, 2002.
[16] C. Lee, H. Abe, T. Hirotsu, and K. Umemura, "Evaluation of the throughput measurement method using connection pairs," WIT '09: Proc. Tenth Workshop on Internet Technology, Japan Society for Software Science and Technology (JSSST), 2009.
[17] V. Paxson and M. Allman, "Computing tcp's retransmission timer," RFC 2988, Internet Engineering Task Force (IETF), 2000.
[18] M. Rak, R. Aversa, B. Martino, and A. Sgueglia, "Web services resilience evaluation using lds load dependent server models.," J. Commun., vol.5, no.1, pp.39–49, 2010.

Appendix: Anomalous Case

We investigate the impact of the packet spacings to the throughput instability at pairs (κ, λ) and (μ, ν). To prove stable network state, we show the CDF of RTT at pairs (κ, λ) and (μ, ν) at Fig. A· 1. Most of values of RTT are close to the mean RTT using ping. It is sufficient to show the stable network. Despite the stable network state, the statistics of network throughput are different from the ideal case at Table 3. We show the CDF of the packet spacing for the judgment at Fig. A· 2. The distribution of the CDF at nodes λ and ν differs from the distribution at nodes β and δ.

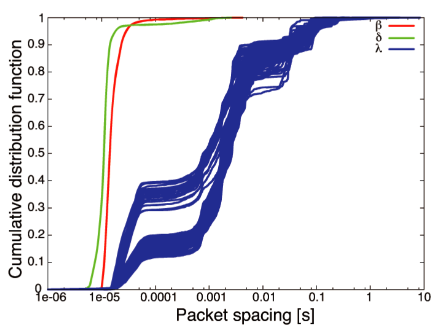


(a) CDF of RTT at κ and λ

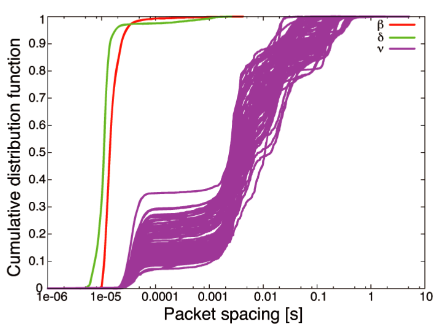


(b) CDF of RTT at μ and ν

Fig. A· 1 CDF of RTT at anomalous case (32 MB).



(a) CDF of packet spacing at β, δ, and λ



(b) CDF of packet spacing at β, δ, and ν

Fig. A· 2 CDF of packet spacing at anomalous case (32 MB).

Chunghan Lee received the B.Eng. from Korea University of Tech. and Edu., Korea in 2007 and the M.Eng. from Toyohashi University of Tech., Japan in 2010. He is a doctoral course student in the Dept. of Electronic and Information Engineering at Toyohashi University of Tech. His current research interests include network measurement and system software for network virtualization areas.

Hirotake Abe received the B.Eng. degree in 1999, the M.Eng. degree in 2001, and the Ph.D. degree in 2004, all from University of Tsukuba, Japan. From 2004 to 2007, he was a research staff of Japan Science and Technology Agency. From 2007 to 2010, he was an Assistant Professor in Toyohashi University of Tech., Japan. He is currently an Assistant Professor in Cybermedia Center, Osaka University, Japan. His research interests include system software, distributed systems and computer security. He received the distinguished paper award from IPSJ in 2005.

Toshio Hirotsu received the Ph.D. degree in computer science from Keio University in 1995. From 1995 to 2004, he worked in NTT Laboratories, Japan. He was in the Dept. of Information and Computer Science at Toyohashi University of Tech. as an Associate Professor from 2004 to 2009. He is currently a Professor of the faculty of the Computer and Information Science at Hosei University. His research interests include system software for Internet and the ubiquitous environment.

Kyoji Umemura received the B.Eng., the M.Eng. and the Ph.D. degrees from the University of Tokyo, Japan in 1981, 1983 and 1991 respectively. Currently he is a Professor in the Dept. of Computer Science and Engineering at Toyohashi University of Tech. His research interests include Information Retrieval, Lisp and Symbolic Computation, Compiler, Operating System and Natural Language Processing.