

석사학위논문

Analysis of Bursty Packet Loss Characteristics
on Underutilized Links

정승화 (鄭昇和)

전자컴퓨터 공학부 (컴퓨터공학)

네트워크 전공

포항공과대학교 일반대학원

2005

Underutilized 링크의 패킷손실에 관한
특징 분석

Analysis of Bursty Packet Loss Characteristics
on Underutilized Links

Analysis of Bursty Packet Loss Characteristics on Underutilized Links

by

Seung-Hwa Chung

Division of Electrical and Computer Engineering

(Computer Science and Engineering)

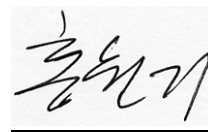
POSTECH

A thesis submitted to the faculty of POSTECH in partial fulfillment of the requirements for the degree of Master of Science in the Division of Electrical and Computer Engineering (Computer Science and Engineering)

Pohang, Korea

December 21, 2005

Approved by



Major Advisor

Analysis of Bursty Packet Loss Characteristics on Underutilized Links

정 승 화

위 논문은 포항공과대학교 전자컴퓨터공학부 (컴퓨터공학) 석사 학위논문으로 학위논문 심사위원회를 통과하였음을 인정함.

2005년 12월 21일

학위논문심사 위원회 위원장 홍원기 (인)

위 원 김 종 (인)

위 원 서영주 (인)

MECE 정 승 화, Seung-Hwa Chung, Analysis of Bursty Packet Loss
20032196 Characteristics on Underutilized Links, Underutilized 링크의 패
 킷손실에 관한 특징 분석, Division of Electrical and Computer
 Engineering (Computer Science and Engineering), 2005, 46P,
 Advisor: J. Won-Ki Hong, Text in English.

ABSTRACT

ISPs and enterprises typically provide sufficient bandwidth according to increasing traffic requirements. However, in the case of QoS-sensitive applications such as VoIP, service quality may not be up to expectations even in underutilized links because of sporadic but non-negligible losses due to traffic bursts. This study aims to detect and analyze packet loss characteristics on underutilized links in an enterprise network environment. We collected packet loss data from a dormitory backbone switch deployed on POSTECH's campus network. To obtain the packet loss information, we fetched data from private SNMP MIB variables of the monitored switch and traffic information using TAP on the monitored link. We analyzed the data and identified parts that are representative of packet loss across various time scales: 10 millisecond, one second, 10 second and one minute. Analysis reveals that packet losses on underutilized link occur by a number of packets in a small time scale and packet losses are affected by a number of bursty packets more than bursty bytes. We also observed other traffic properties such as packet size distribution and flows for the packet loss on underutilized link. Our multi-resolution analysis shows that links that appear underutilized at coarse time granularity, often the case in production network monitoring, reveal burstiness and resulting losses at fine time granularity relevant for VoIP and other QoS-sensitive traffic.

Table of Contents

1.INTRODUCTION	1
2.RELATED WORK.....	3
2.1. Traffic Monitor using SNMP	4
2.1.1. Limitation of SNMP.....	5
2.2. Traffic Monitor using TAP on link.....	7
2.2.1. Libpcap & SOCKET_PACKET Performance.....	7
3.PACKET LOSS DETECTION & TRAFFIC MONITORING	15
3.1. Packet Loss Detection	15
3.2. Traffic Monitoring.....	16
3.3. Packet Loss & Traffic Monitoring Method	19
4.TRAFFIC DATA COLLECTION.....	20
5.ANALYSIS OF PACKET LOSS AND IP TRAFFIC	24
5.1. Bursty Traffic Analysis	24
5.2. Bursty Traffic in Small Time Scale.....	30
5.3. Packet Size Distribution	35
5.4. Flow Analysis	38
5.4.1. Flow Life Time.....	39
5.4.2. One-Tuple Based Flow.....	41
6.CONCLUDING REMARKS & FUTURE WORK	43
REFERENCES.....	45

Table of Figures

Figure 1. APIs Testing Module Design	8
Figure 2. Libpcap 70 Byte Packets Capture Performance.....	9
Figure 3. SOCKET_PACKET 70 Byte Packets Capture Performance	10
Figure 4. Libpcap 1500 Byte Packets Capture Performance.....	11
Figure 5. SOCKET_PACKET 1500 Byte Packets Capture Performance	12
Figure 6. Linux Network Kernel Overview	13
Figure 7. Traffic Burst Detection Module Design.....	17
Figure 8. Overview of Traffic Monitoring Modules	18
Figure 9. POSTECH's Campus Network Overview	21
Figure 10. POSTECH intranet infrastructure for Dormitories	22
Figure 11. Experimental Environment Overview	22
Figure 12. Incoming Packet Distribution	25
Figure 13. Incoming Byte Distribution	25
Figure 14. Packet Loss Distribution.....	27
Figure 15. Packet Loss Distribution of All Ports.....	29
Figure 16. CPU Utilization	29
Figure 17. Incoming Burst Packet Distribution without Packet Loss	31
Figure 18. Incoming Burst Packet Distribution with Packet Losses	31
Figure 19. Incoming Burst Byte Distribution without Packet Loss	33
Figure 20. Incoming Burst Byte Distribution with Packet Losses	33
Figure 21. Interval Time between passing Packets with Packet Losses.....	34
Figure 22. Interval Time between passing Packets without Packet Loss.....	35

Figure 23. Packet Size Distribution.....	36
Figure 24. Packet Size Distribution with Packet Losses	37
Figure 25. Packet Size Distribution without Packet Loss	37
Figure 26. Flow Distribution and Packet Loss Distribution.....	38
Figure 27. TCP Flow Distribution.....	40
Figure 28. UDP Flow Distribution	40
Figure 29. One-tuple based flow at 1 second granularity.....	42
Figure 30. One-tuple based flow at 10 millisecond granularity	42

List of Tables

Table 1. SNMP standard MIB II variables	4
Table 2. SNMP Data from DPNM Lab switch	6
Table 3. System Resource for Test	8
Table 4. Cisco Enterprise MIB variables.....	15
Table 5. Implemented Module List in Traffic Monitor System.....	19
Table 6. Incoming Packet Statistics.....	26
Table 7. Incoming Bytes Statistics	26
Table 8. Incoming Burst Packet Statistics	30
Table 9. Incoming Burst Byte Statistics	32

1. Introduction

Today, the number of Internet users is continually increasing, along with the number of networked applications. These newly emerging network-based applications including VoIP, teleconferencing, streaming media, peer-to-peer, and games generate a significant amount of traffic. This trend is accelerating because of broadband connections and improved PC performance. Increased traffic load underlies many performance and security related problems in both the Internet and enterprise networks.

Most real networks, including backbone, enterprise, and some access networks, employ overprovisioning to mitigate performance problems such as packet loss, delay, and jitter for QoS-sensitive applications. To protect against quality degradation at bottleneck segments in large, heterogeneous IP internets, priority scheduling is used to shield QoS-sensitive traffic from best-effort traffic. For example, Cisco routers use low latency queue (LLQ) in IOS as a basic building block to support VoIP. Unlike telephony, however, where uniform standards enable the allocation of end-to-end bandwidth across multiple providers and equipment vendors via TDM channels, IP internet works crucially rely on overprovisioning to facilitate end-to-end QoS.

An important component of traffic management for resource provisioning and network planning is traffic monitoring. Some ISPs are overprovisioning their link bandwidth by observing traffic monitor systems such as NetFlow [1], MRTG [2] and NG-MON [3]. Those systems monitor traffic with large time interval and overprovisioned ISPs do not have satisfied result on QoS sensitive applications. Because their monitoring systems cannot detect bursty traffic on a small time

scale [4]; their coarse time granularity is limited to five minute and one minute aggregates in the case of MRTG and NG-MON, respectively. Due to averaging over large time intervals we may see links being underutilized (e.g., 20% or less) when, in fact, at the time granularity smaller than a second, critical for assuring toll quality VoIP, traffic spikes and packet losses are present leading to unacceptable service violations. This is especially relevant given the self-similar nature of Internet traffic [5].

In this paper, we will prove that packet loss exists on underutilized links and we will analyze packet loss with various traffic parameters (number of packets, packet size distribution and flows, etc) observed by our traffic monitoring system.

Knowing the characteristics of packet loss, we expect this study can assist in designing new applications or router structures that are prepared to handle unexpected packet loss.

The organization of this paper is as follows. Relative work is presented in Section 2 and our packet loss and traffic monitor system is described in Section 3. Section 4 describes the traffic data collection and experimental environment. In Section 5, we give an analysis of packet loss and IP traffic. Finally, concluding remarks are given and possible future work is discussed in Section 6.

2. Related Work

People take no packet loss on underutilized link as a matter of course on underutilization link. Not many studies in the area of packet loss on underutilized links are available. The following summarizes research related to packet loss is relevant to the present study.

Papagiannaki et al. [6] presented a characterization of congestion in the Sprint IP backbone network. They analyzed link utilization at various time scales (millisecond level) to measure the frequency and duration of micro congestion. While they detect traffic bursts and their duration, they do not mention the packet loss that occurred during these times. This study did not provide various traffic parameters except burst, and no information of the packet loss. Hence, further work is needed on this topic. In our paper, we will provide the packet loss characteristics with various traffic parameters.

Hall et al. [7] analyze the effect of early packet loss on web traffic and its download time. They discovered that the TCP SYN packet loss causes higher latency in web page downloads than other types of packet losses. This work concentrates on web traffic, and showed that a small amount of packet loss can contribute to serious delays (second level). The packet loss on underutilized link dose not occurs often but these small packet losses can be important factor for QoS-sensitive traffic.

Mochalski et al. [8] studied changes in traffic pattern relative to different points of observation using TAP in the network and investigated the contributing factors to the changes observed. They measured the delay across the router and firewall and tried to relate the delay to packet loss, concentrating on analyzing

packet loss using delay. We try to analyze packet loss using a range of parameters, such as the number of incoming and outgoing packets, bandwidth, and the number of flows [9]

2.1. Traffic Monitor using SNMP

Traditional way of monitoring system is using SNMP [10]. MRTG is one of the most popular monitoring systems using SNMP. A SNMP agent is used to obtain traffic bandwidth from SNMP enabled routers and switches. From the supported SNMP MIB II, we selected and fetched data for four MIB variables, ifInUcastPkts, ifOutUcastPkts, ifInOctets, and ifOutOctets. The selected SNMP MIB II variables are described in Table 1.

Object	OID	Description
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11	The number of subnetwork-unicast packet delivered to a higher-layer protocol
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17	The total number of packets that higher-level protocols requested be transmitted to a subnetwork-unicast address, including those that were discarded or not sent
ifInOctets	1.3.6.1.2.1.2.2.1.10	The total number of octets receive on the interface, including framing characters
ifOutOctets	1.3.6.1.2.1.2.2.1.16	The total number of octets transmitted out of the interface, including framing characters

Table 1. SNMP standard MIB II variables

2.1.1. Limitation of SNMP

We tried to detect packet loss by comparing the incoming and outgoing packet counters. However, the loss information obtained is inaccurate for the following reasons.

- ✓ Some packets are destined to the router. Therefore, there is no outgoing packet for such packets.
- ✓ Some packets are generated by the router. Therefore, there is no incoming packet for such packets.
- ✓ Some packets are broadcasted by the router. This causes a large difference between the incoming and outgoing packet counters and confuses the loss count.

The above problems can not be avoided using SNMP MIB II variables, and standard MIB II does not offer traffic characteristics that we are considering for packet loss.

We also observed inaccuracies in the data collected from some of the devices using SNMP, which resulted from time lags in the devices' counter update interval. A part of the data from the DNMP Lab switch (port 24) that illustrates the counter inaccuracies is shown in Table 2.

Time in Second	In Packets	Out Packets	In Bytes	Out Bytes
3:23:20 pm	0	0	64	0
3:23:21 pm	0	0	0	0
3:23:22 pm	400	522	79669	454646
3:23:23 pm	0	0	64	609046
3:23:24 pm	437	611	67670	0
3:23:25 pm	0	0	64	0

Table 2. SNMP Data from DPNM Lab switch

The columns of the table show MIB polling time, incoming packets, outgoing packets, incoming bytes, outgoing bytes and CPU utilization, respectively. The data indicates the presence of events when the number of incoming packets was zero for a period of one second, which we know is not true because the DPNM Lab switch continually received packets during the specified time interval. The discrepancy is caused by MIB counters not being updated immediately after a packet arrives. A router/switch's main priority is forwarding packets, which can cause counter updates to be delayed, especially when traffic load is heavy. To find the minimum time interval that can be used to reliably poll the selected devices using SNMP, we increased the polling interval from one second to two seconds and so on. We found that with the polling interval of 10 seconds, the data obtained is sufficiently accurate on our experimental environment.

2.2. Traffic Monitor using TAP on link

Ten seconds of traffic monitoring time interval is still not a satisfied small time interval to detect traffic burst. In case of MRTG system, it offers 5 minutes of traffic monitoring time interval and traffic burst will not be detected due to averaging over large time intervals. Traffic monitor using TAP can solve those time interval problem, because using TAP on link means capturing passing packets at real time. Most traffic monitor systems that use TAP are implemented with packet capture APIs such as Libpcap and SOCKET_PACKET, etc.

2.2.1. Libpcap & SOCKET_PACKET Performance

We tested Libpcap and SOCKET_PACKET that are most popularly used on traffic monitor system. We used a traffic generator (SmartBits 600) to test by sending custom packets. Table 3 shows the system resource and device that we used during the test, and figure 1 shows how the packet capture module designed. We design the module to minimize the part for packet analysis (only IP packets check and packet counter are implemented in test module) to find out packet capture performance of APIs. Next shows the performance test result.

SmartBits 600	2 x 1Gigabit packet generator (TX & RX)
Linux Monitoring System	CPU: Single Intel Pentium IV 2.8GHz
	Memory: 512MByte
	eth0: 3 Com (100Mbit NIC)
	eth1: Intel Ethernet Pro 1000 (1Gbit NIC)
	eth2: Intel Ethernet Pro 1000 (1Gbit NIC)
	OS: RedHat Linux 9.0 (Kernel 2.4.20-8)

Table 3. System Resource for Test

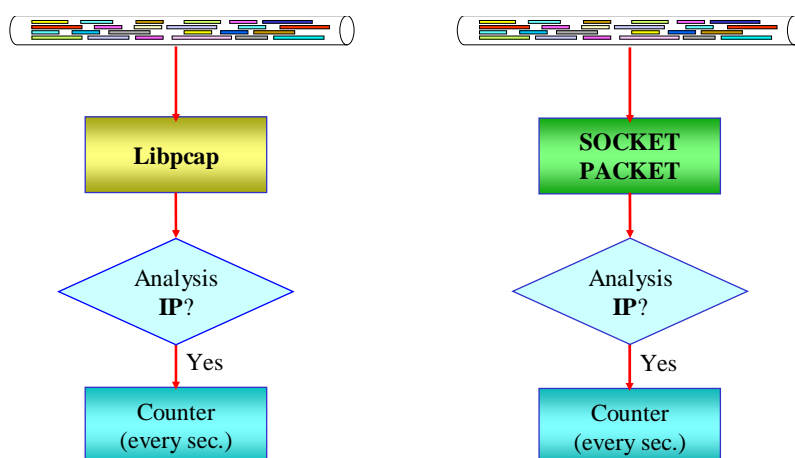


Figure 1. APIs Testing Module Design

The traffic generator (SmartBits 600) was set to send 70 byte IP packets, and Libpcap was set to capture only IP headers. As you can see in figure 2, our first test generated 1000 packets per second for capture. At 1000 packets per second (pps) we could not find any packet loss on the libpcap test module. We increased the traffic generator's pps incrementally from 1000 up to 150000, and we found that around 50000 pps libpcap began to experience some packet losses in the test

module. 150000 pps corresponds to 84 Mbits per second, which is less than the full bandwidth of a 100 Mbits link. At 150000 pps the link is underutilized, in figure 2, we can see a high number of packet losses (60000) and high CPU usage (99%).

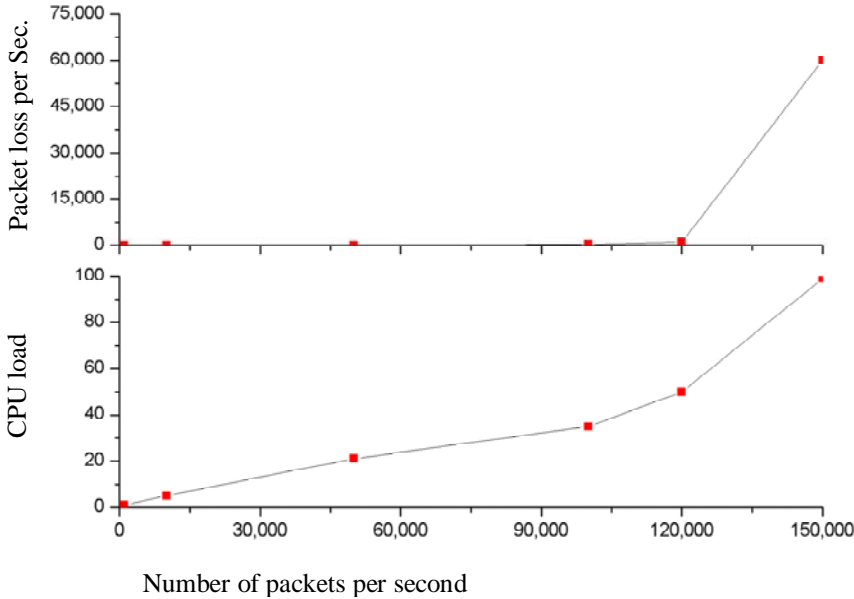


Figure 2. Libpcap 70 Byte Packets Capture Performance

Same test was performed using SOCKET_PACKET. The Traffic generator (SmartBits 600) was set to send 70 byte IP packets, and SOCKET_PACKET was set to capture only IP headers. As you can see in figure 3, our first test generated 1000 pps for capture. At 1000 pps, we could not find any packet loss on the SOCKET_PACKET test module. We increased the traffic generator's pps incrementally from 1000 up to 150000, and we found that around 50000 pps the

SOCKET_PACKET experienced some packet losses in the test module. As stated earlier, at 150000 pps the link is not fully utilized, in table 4, we can see a high number of packet losses (27000) and high CPU usage (97%)

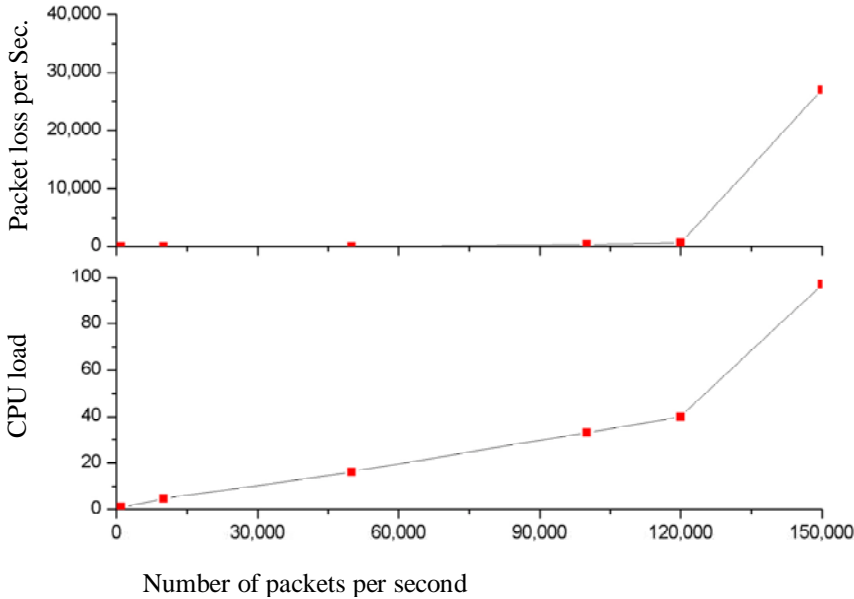


Figure 3. SOCKET_PACKET 70 Byte Packets Capture Performance

This time we set the traffic generator (SmartBits 600) to send 1500 byte IP packets, and Libpcap was set to capture only IP headers. As you can see in figure 4, our first test generated 1000 pps for capture. Again at 1000 pps, we could not find any packet loss on the libpcap test module. We increased the traffic generator’s pps incrementally from 1000 up to 80000. At 10000 pps the libpcap test module had no packet losses. 10000 pps corresponds to 120 Mbits per second,

which is more than the full bandwidth of a 100 Mbits link. So in the case of 1500 Byte packets or little less size of packets, we can say libpcap can capture packets on a fully utilized link without loss. However, real time internet traffic is composed of various sizes of packets. We cannot assuredly say that libpcap can handle 100 Mbits Link.

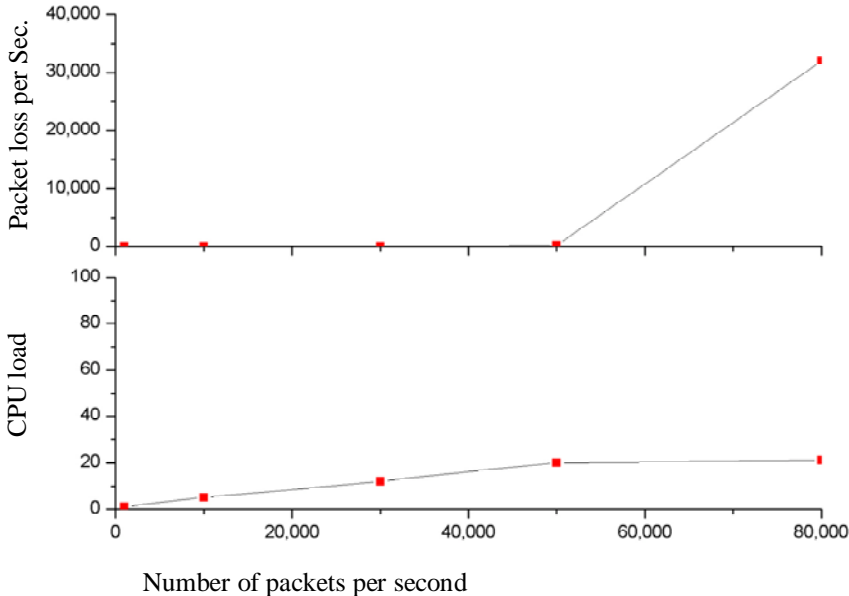


Figure 4. Libpcap 1500 Byte Packets Capture Performance

Same test was performed using SOCKET_PACKET. The Traffic generator (SmartBits 600) was set to send 1500 byte IP packets, and SOCKET_PACKET again was set to capture only IP headers. As you can see in figure 5, our first test generated 1000 pps for capture. Again at 1000 pps, we could not find any loss on SOCKET_PACKET test module. We increased traffic generator's pps

incrementally from 1000 up to 80000, and we found that at 10000 pps the SOCKET_PACKET test module had no packet losses. So in the case of 1500 byte packets or lesser size packets, we can say SOCKET_PACKET can capture packets on a fully utilized link without loss. As we mentioned earlier, real time internet traffic is composed of various sized packets. We also cannot say assuredly that SOCKET_PACKET can handle a 100 Mbits Link.

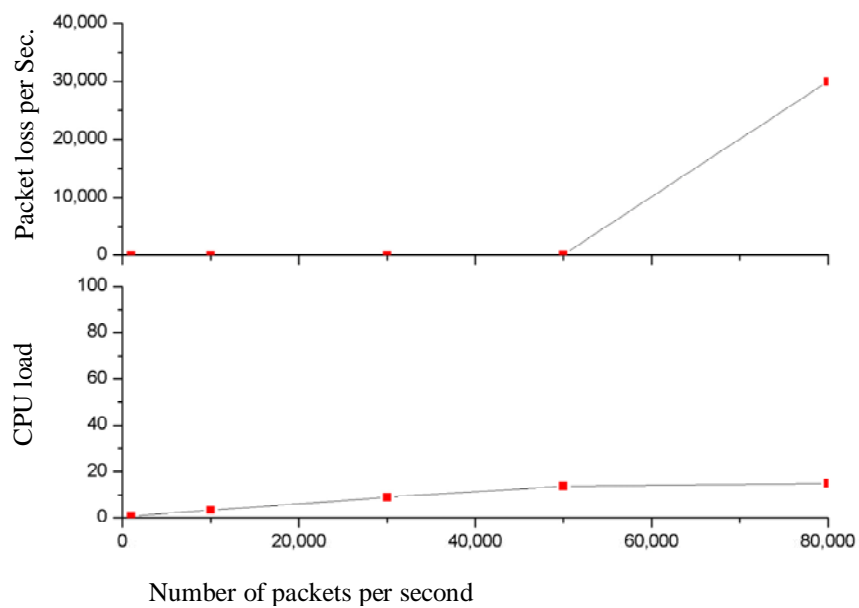


Figure 5. SOCKET_PACKET 1500 Byte Packets Capture Performance

By the test result, SOCKET_PACKET showed a slightly better performance than Libpcap in overall packet processing. The problem, however, is that both APIs could not handle the 100Mbps link when link bandwidth is fully utilized by 70 byte packets in our system. In real internet traffic, packets passing link will not

be always size of 70 byte. Our research is on underutilized links so the above APIs we tested may satisfy for a 100 Mbps link, but in this paper we collect traffic data from a one Gbps link using TAP. In case of one Gbps link, packet capture APIs with our system resource did not show good enough performance.

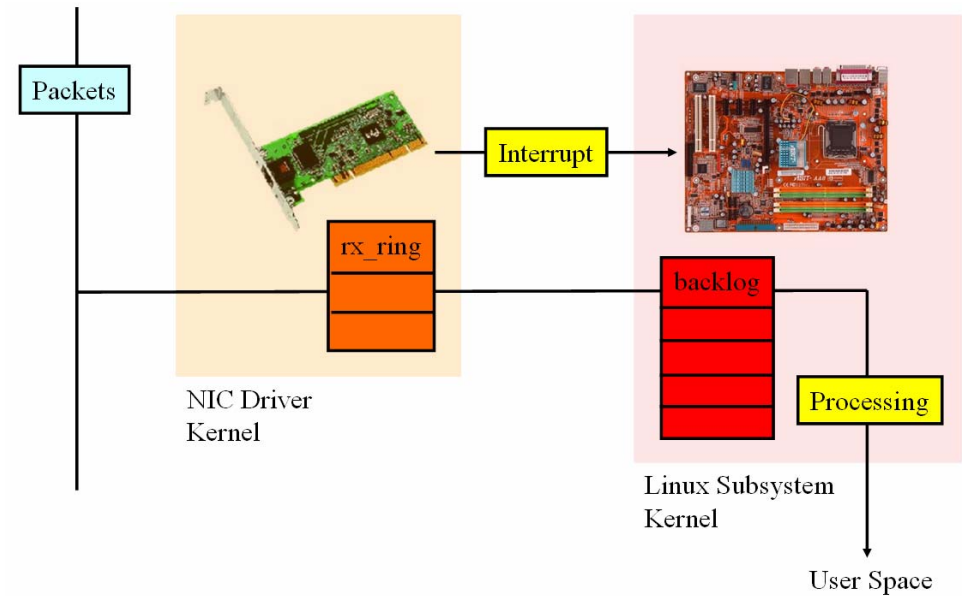


Figure 6. Linux Network Kernel Overview

Figure 6 illustrates the overview of the linux network kernel, and by observing packet process in kernel level we could find where the packet capture APIs have packet losses. The following illustrates how packets flow through the kernel of a linux system.

1. NIC captures a packet; it stores a packet in a rx_ring on the NIC.
2. If the rx_ring is not empty, the NIC gets a packet from the rx_ring and sends to linux network subsystem.

(When the NIC hands-off a packet to the linux network subsystem, the NIC sends an interrupt signal to linux system kernel.)
3. Linux network subsystem stores the received packet in a backlog.
4. If the backlog is not empty, the linux kernel gets a packet from the backlog and it is processed.
5. The linux kernel sends the processed packet information to the user space.

Packet drops occur on our test system, when the packet passes from the rx_ring (Buffer inside NIC) to the backlog (Buffer in Linux System), because the linux network subsystem kernel could not manage all packets handed from NIC, so backlog was often over flowed. Thos packet capture APIs obtain the packet information after packets are processed from the backlog, so APIs do not obtain packet information that dropped between the rx_ring and the backlog. The linux subsystem kernel could not handle a fully utilized (by 70 byte packets) 100 Mbits link without packet loss with our system resource.

3. Packet Loss Detection & Traffic Monitoring

3.1. Packet Loss Detection

Related work, we explained using SNMP standard MIB can not monitor packet losses, so we decided to use Cisco enterprise MIB variables [11] that provide packet loss and CPU usage information: `locIfInputQueueDrops`, `locIfOutputQueueDrops` and `cpuLoad`. The selected enterprise MIB variables are described in Table 4.

Object	OID	Description
<code>locIfInputQueueDrops</code>	1.3.6.1.4.1.9.2.2.1.1.26	The number of packets dropped because the input queue was full
<code>locIfOutputQueueDrops</code>	1.3.6.1.4.1.9.2.2.1.1.27	The number of packets dropped because the output queue was full
<code>cpuLoad</code>	1.3.6.1.4.1.9.2.1.56	CPU Utilization (5 sec avg.)

Table 4. Cisco Enterprise MIB variables

In Cisco document [12], it is written: Each interface owns an input queue onto which incoming packets are placed to await processing by the Routing Processor (RP). Frequently, the rate of incoming packets placed on the input queue exceeds the rate at which the RP can process the packets.

The information of dropped packet number, because the interface input queue was full, is updated in `locIfInputQueueDrops` MIB variable in accumulated order, and the packet number, because the interface output queue was full, is updated in `locIfOutputQueueDrops` MIB variable in accumulated order.

3.2. Traffic Monitoring

We decide to monitor link using TAP because MIB has the limitations of polling time interval and information of traffic. Detecting traffic with a short time granularity and obtaining detail traffic information, TAP or mirroring port are required to monitor a link. Mirroring port is easy to use but this method makes a router work on overload. Since router overload effects packet loss, we decided not to use the mirroring function of the router. In our study, to monitor traffic we use TAP on a link and implement monitor system. Related work, we tested the performance of packet capture APIs (Libpcap and SOCKET_PACKET), and those APIs showed not enough performance to detect burst packet on 1 Gbps link. Our traffic monitor system requires the performance that can manage very short time granularity burst packets, even router or switch can not handle and appear to have packet losses. So we decided to implement a traffic burst detection module in the kernel to detect burst traffic without loss even in a one Gbps link.

We implemented a traffic burst detection module in the NIC driver to detect burst traffic as you can see in figure 7. When the packet arrives to NIC the traffic burst detection module obtains the size of the packet. And when an interrupt signal occurs in the NIC, the module writes the packet information to the linux proc file system placed in kernel memory. Counter is updated every 10 millisecond with the information in proc file system.

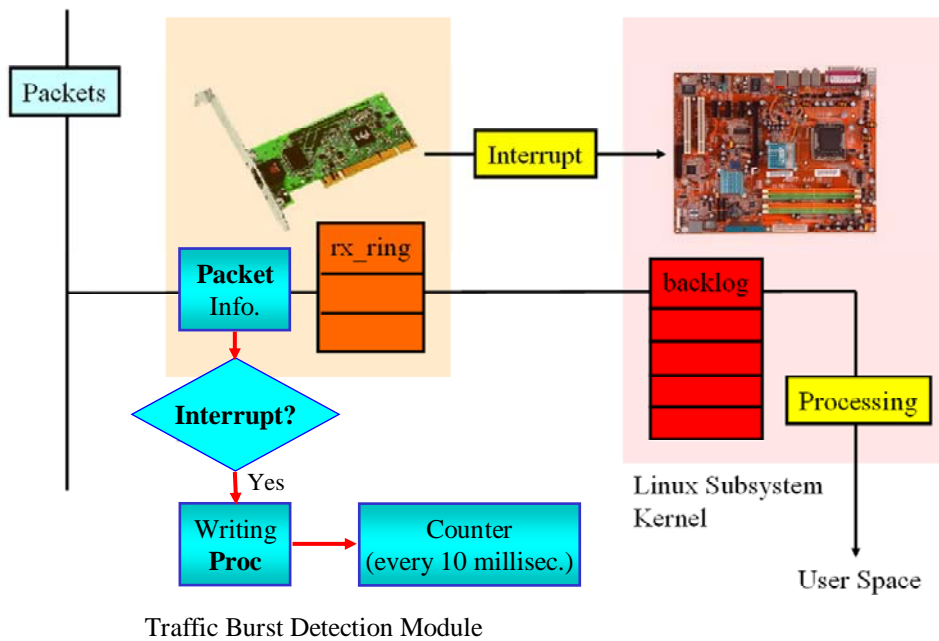


Figure 7. Traffic Burst Detection Module Design

By handling packets in the NIC driver kernel, we could increase packet capture performance. The information obtained from the kernel contains only two values (number of packet & size of packet) in order to maximize the performance. These two parameters can give us traffic bursty information in a short time granularity.

We tested the performance by sending 70 byte packets and there was no packet loss up to 1,000,000 pps which corresponds to 560 Mbits per second. When pps was higher than 1,000,000 our traffic burst detection module in the kernel started to suffer packet loss. Sending 1500 byte packets, the traffic burst detection module in the kernel could handle a fully utilized 1 Gbps link without

any loss. We were satisfied with this result even though it could not manage more than 560 Mbits (more than 50% of utilization) in 1 Gbps link in terms of 70 byte packets without loss. Because in real internet traffic, the size of packets passing through a link will not always be 70 Byte; and our research is on underutilized (around 2% of utilization) links, so this test results satisfy our study criteria of packet loss on underutilized links. We implemented this kernel modified traffic burst detection module on a traffic monitoring system to detect burst traffic that is required to be very accurate in a short time granularity. Additionally, we implemented a packet trace module and flow module to analyze packet loss characteristics among many different traffic parameters as shown in table 5 and figure 8 illustrates the overview of modules in traffic monitor system.

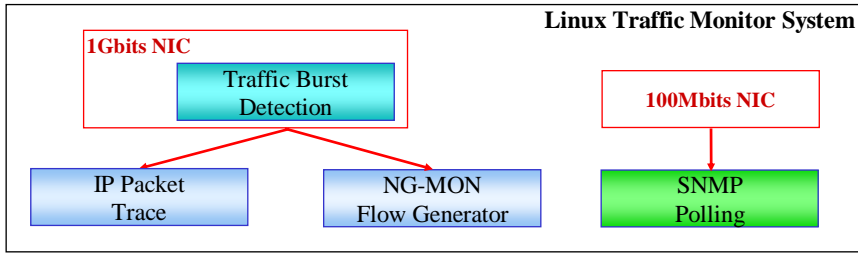


Figure 8. Overview of Traffic Monitoring Modules

Module Name	Module Description
Traffic Burst Detection Module	This module detect traffic burst (number of packet, size of packet) at short time granularity.
SNMP Polling Module	This module polls Cisco private MIB and Standard MIB II values.
IP Packet Trace Module	This module capture all IP packets and save their header with time stamp.
NG-MON Flow Generator Module	This module generates packets into flows and used for detail flow analysis.

Table 5. Implemented Module List in Traffic Monitor System

3.3. Packet Loss & Traffic Monitoring Method

In this paper, we implemented a monitoring system that can detect traffic burst accurately in a small time granularity (10 milliseconds). Several of the modules in the system are shown in table 5. We monitored traffic on the link using TAP and at the same time we monitored packet losses by polling Cisco private MIBs using SNMP. Bursty packet loss and traffic detection methods on underutilized links are described below.

1. If there is a packet loss, we can obtain the time the packet loss occurred in accuracy of 10 seconds.
2. At the same time the traffic on the link is monitored by our implemented system, we also collect traffic data in ten second intervals to verify packet loss in the data sets.
3. We analyze data sets obtained by the step above in offline, and show bursty packet loss characteristics on underutilized links.

4. Traffic Data Collection

SNMP agents are running on various network devices deployed in the campus network. POSTECH's campus network is comprised of a gigabit Ethernet backbone, which, in turn, is composed of two Cisco IP routers, two core backbone switches, dozens of gigabit building backbone switches, and hundreds of 100Mbps switches and hubs that are deployed inside the buildings, as shown in figure 9. We tried to find the link that continuously underutilized and has the traffic that composed of many internet applications. Our campus internet access links are not underutilized and we keep traced the link to find the most suitable place to observe packet losses on underutilized link. We found the link that satisfied above conditions is placed on campus dormitory network. This link is the edge of 1 Gbps links in POSTECH's campus network and continuously underutilized. Because this link is placed in dormitory network, many students are connected and use the internet. The traffic generated by the students is representative of traffic that ISPs handle from their users.

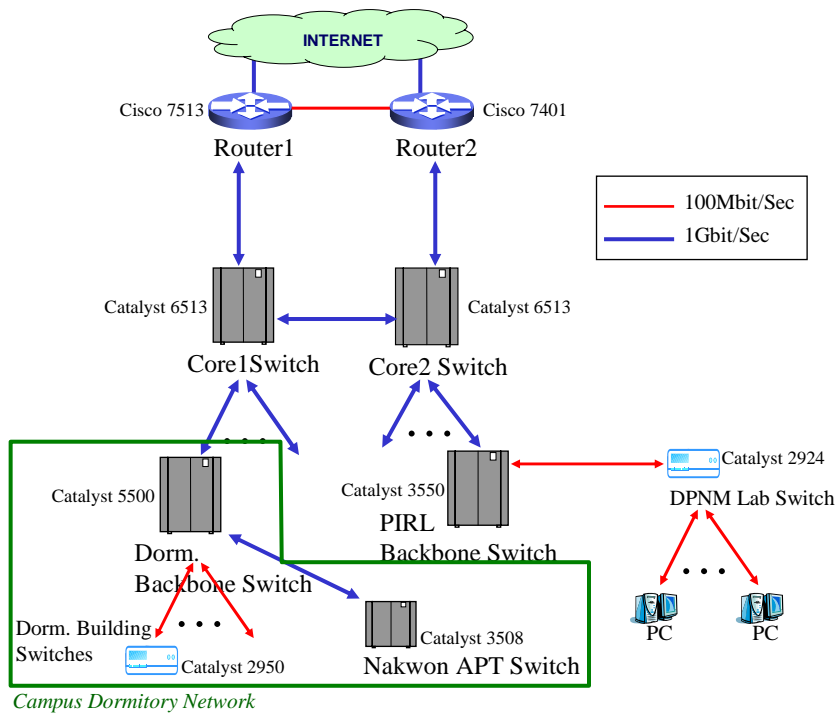


Figure 9. POSTECH's Campus Network Overview

We decided to monitor packet loss from the dormitory backbone switch (a Catalyst 5500) that is placed next to the Core switches (Catalyst 6513). This dormitory backbone switch is connected with many sub-dormitory switches as you can see in figure 10. We monitored all links between the dormitory backbone switch and sub-dormitory switches and found the link between sub-dormitory switch (Catalyst 3508) for “Nakwon APT” and dormitory backbone switch (Catalyst 5500) was the most suitable for our study. This link was a good choice because it was underutilized and it showed the occurrence of steady packet losses.

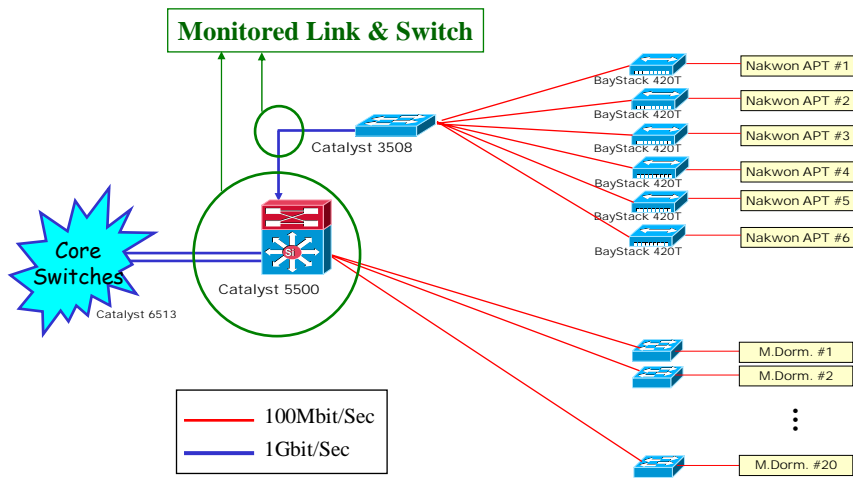


Figure 10. POSTECH intranet infrastructure for Dormitories

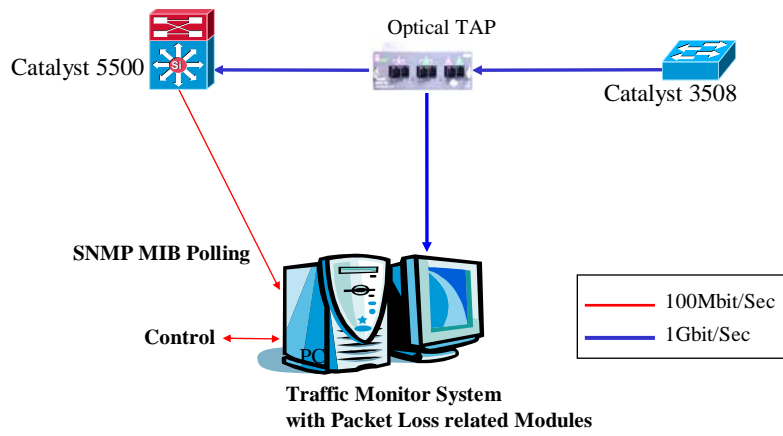


Figure 11. Experimental Environment Overview

We installed an optical TAP on the link that connects the dormitory backbone switch (Catalyst 5500) and sub-dormitory switch (Catalyst 3508) as you can see in Figure 11. The dormitory backbone switch offers Cisco private MIBs, so we could obtain packet loss data using SNMP (a SNMP polling module was implemented in the traffic monitor system.).

We monitored one-direction traffic that flows from the sub-dormitory switch to the dormitory backbone switch. Because interface output queue drop (locIfOutputQueueDrops) showed no packet loss and traffic properties on both direction in/out are almost same.

We monitored only one link (not all links of the dormitory backbone switch), and monitoring one link is enough to obtain packet loss characteristics, because we analyze the traffic of the link when the packet loss occurs only on monitored port. Then we know packet losses are affected only by the traffic that we monitored, not by other ports traffic.

5. Analysis of Packet Loss and IP Traffic

This section presents an analysis of IP traffic and packet loss. We collected data for one week, 2004.11.23 2:00 pm to 2004.11.30 3:00 pm from the specified switch and link. We collected data passively using TAP and polling MIB values at a 10 second granularity and aggregated to yield one minute and five minute data. We focused on the parts that clearly illustrated packet loss.

5.1. Bursty Traffic Analysis

Figure 12 and 13 illustrate the distribution of incoming packets and incoming bytes over the five days period at 10 second, one minute and five minute measurement granularity. This data is collected from the Dormitory backbone switch 1 Gbps port, which is connected to sub-dormitory switch for “Nakwon APT” and lightly utilized. Overall average utilization of the link is around 5 percent. We observed that the curve of incoming packets and incoming bytes with five minute average values is fairly smooth. The curves with one minute average values clearly show bursty traffic and wider distribution than the five minute measurement curve. Whereas, the curves with 10 second measurement values show even more bursty traffic with high peaks and wider distribution than the one minute measurement curve due to these traffic bursts. These graphs indicate that burstiness in traffic exist in the underutilized link at the small time scale of 10 second.

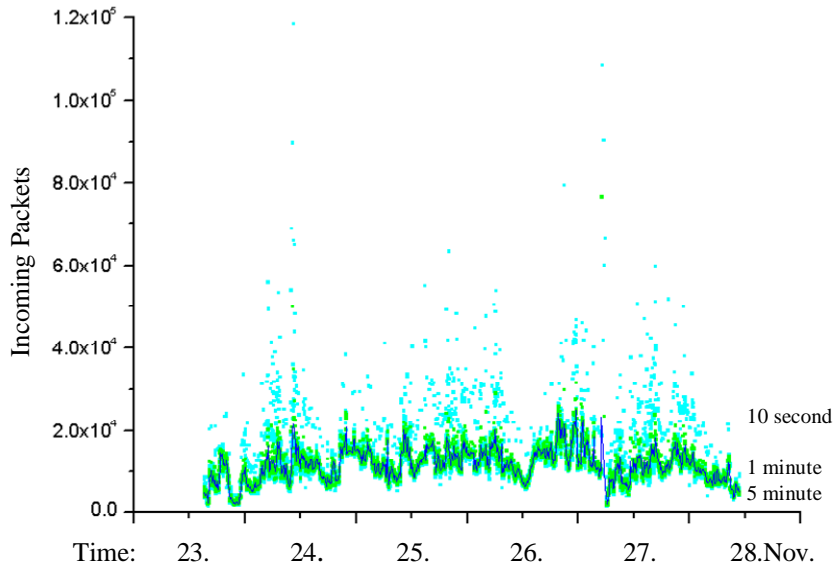


Figure 12. Incoming Packet Distribution

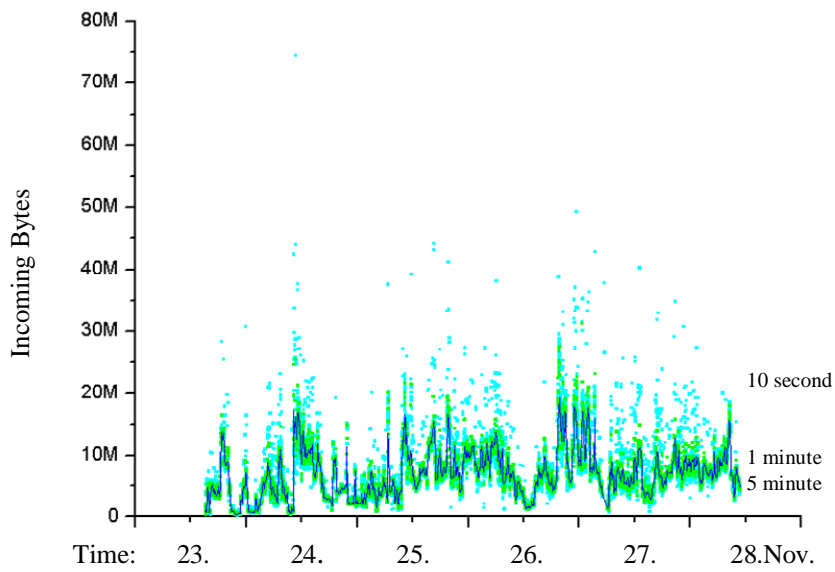


Figure 13. Incoming Byte Distribution

Time Interval	Min.	Max.	Mean	Standard Dev.	Standard Dev. divided by Avg.
10 second	763	766745	10488	6855	0.65
1 minute	1254	45552	10438	4764	0.46
5 minute	1485	33586	10413	4448	0.43

Table 6. Incoming Packet Statistics

Time Interval	Min.	Max.	Mean	Standard Dev.	Standard Dev. divided by Avg.
10 second	112235	95294848	6424751	4578390	0.71
1 minute	160218	49847286	6416863	4257214	0.66
5 minute	285986	45501012	6438273	4144063	0.64

Table 7. Incoming Bytes Statistics

Table 6 and 7 respectively show the statistics for incoming packets and bytes. From the tables we can observe that the mean value of incoming packets and bytes is similar for all time (10-second, 1-minute and 5-minute) scales. On the other hand, standard deviation shows that the distribution of packets and bytes gets wider as the time granularity shortens.

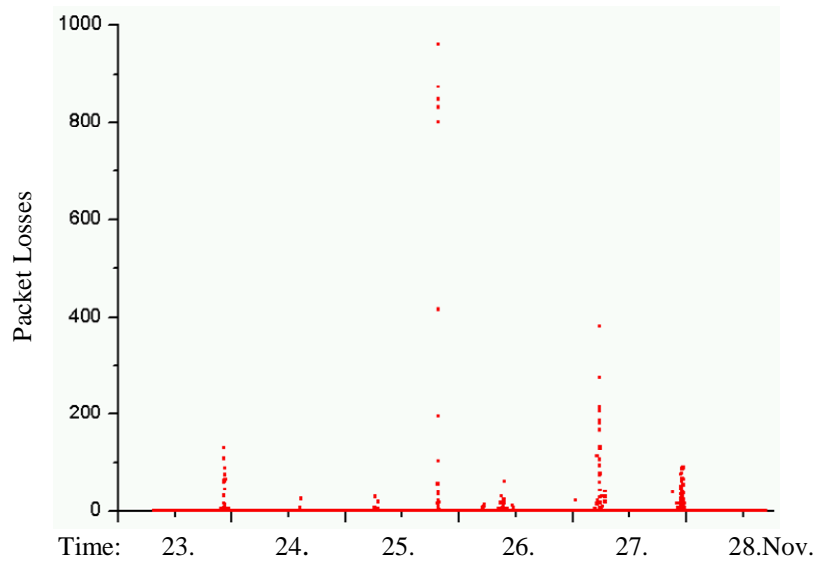


Figure 14. Packet Loss Distribution

Figure 14 and 15 illustrate the distribution of packet loss on the dormitory backbone switch over the seven days and the three days period at 10 second measurement granularity. In figure 14, the graph shows that the packet loss occurred on a port which is connected to the sub-dormitory switch for “Nakwon APT.” during the link on this port was underutilized. In figure 15, the upper graph shows the total packet loss that occurred on the dormitory backbone switch except for the port that is connected to the sub-dormitory switch for “Nakwon APT.”

We tried to analyze traffic data when only our experimental port has the packet loss because we wanted to analyze the packet loss characteristics unaffected by traffic of other ports. In the lower graph of figure 15, we marked a specific time: 2004.11.23 23:43:40. We analyzed traffic properties in this time to discover packet loss characteristics. Figure 16 illustrates CPU utilization over the same period of time in figure 15. From the graph in figure 16, we observe that many packet losses occur when CPU utilization is high (Some CPU utilization peaks matches with the peaks of packet losses) but we can also observe that packet losses occur even when CPU utilization is not high (around 50 percent). There can be a CPU usage burst and hidden; because CPU usage that we obtained is 5 second averaged value.

Figure 13 illustrates monitored link bandwidth over the seven days period. During this period the link bandwidth was continuously underutilized, and as you can see in figure 14, there were packet losses. It proves there are packet losses on the underutilized link, but we do not know the packet loss characteristics on underutilized link. Following analysis will show the traffic characteristics for the packet loss.

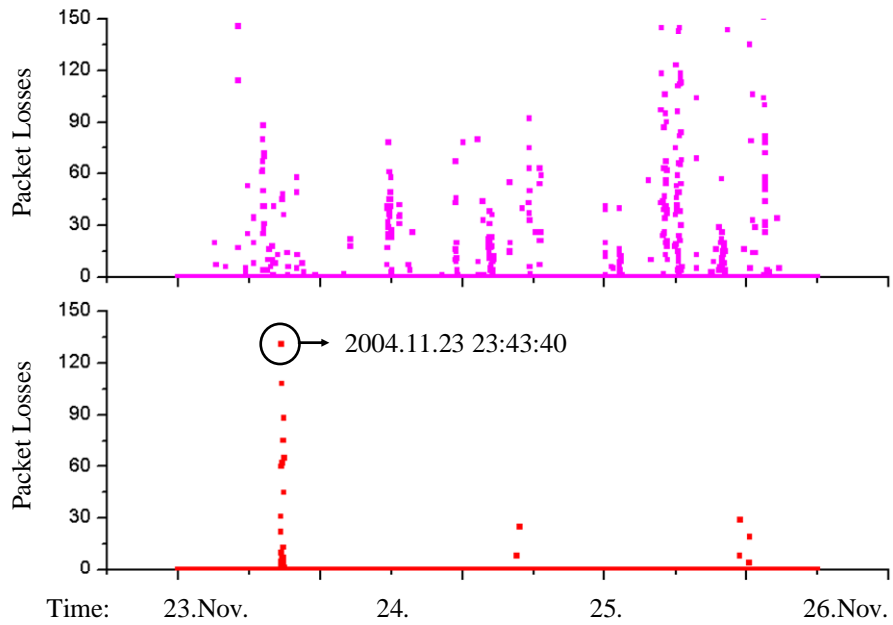


Figure 15. Packet Loss Distribution of All Ports

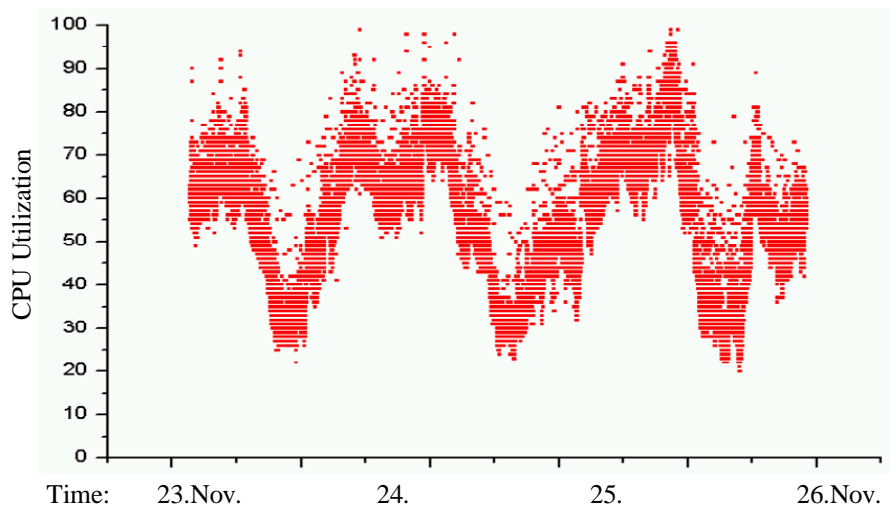


Figure 16. CPU Utilization

5.2. Bursty Traffic in Small Time Scale

The traffic monitoring system collected bursty traffic data from the link (between the Dormitory backbone switch and sub-dormitory switch for “Nakwon APT”) using TAP in a time scale of 10 milliseconds. Figures 17, 18, 19 and 20 illustrate the bursty traffic. Figure 17 shows the bursty packets when there is no packet loss detected and Figure 18 shows the bursty packets when packet losses are detected. Figure 17 and 18 show the different distribution. We can see when there are lost packets the packet distribution shows more burst than when there is no packet loss.

Table 8 respectively shows the statistics for incoming bursty packets. From the table we can observe that the mean value of incoming bursty packets without packet loss and with packet loss is similar for both time (Mean value of incoming bursty packets without packet loss is little higher.) scales. On the other hand, standard deviation shows that the distribution of bursty packets gets wider when there is the packet loss.

Packet Loss?	Min.	Max.	Mean	Standard Dev.	Standard Dev. divided by Avg.
No	1	32	11	4	0.36
Yes	0	196	8	15	1.87

Table 8. Incoming Burst Packet Statistics

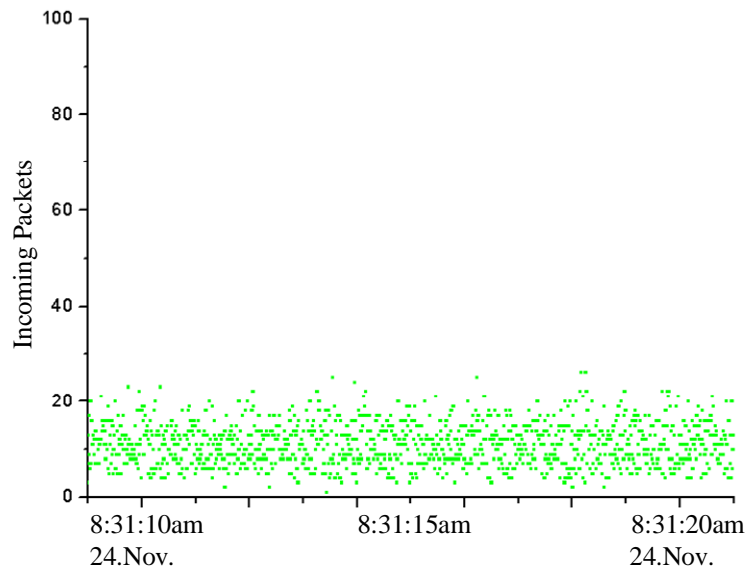


Figure 17. Incoming Burst Packet Distribution without Packet Loss

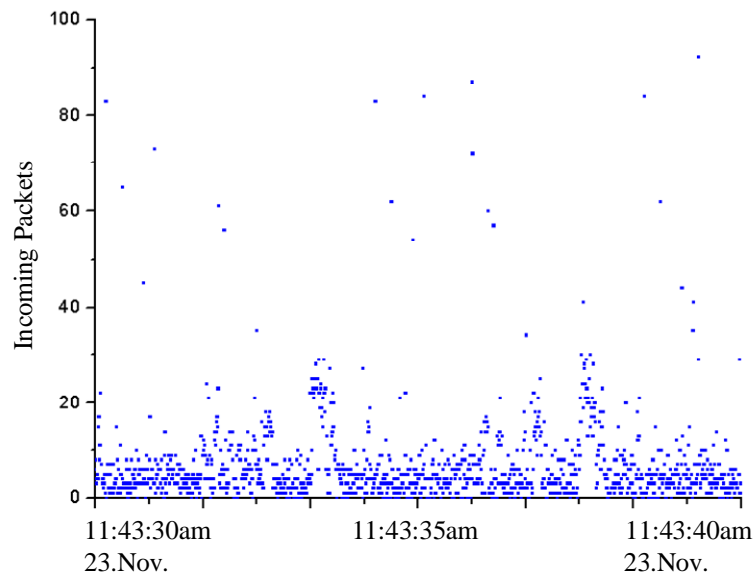


Figure 18. Incoming Burst Packet Distribution with Packet Losses

Figure 19 shows the bursty bytes (link bandwidth in granularity of 10 milliseconds) when there is no packet loss detected and Figure 20 shows the bursty bytes when packet losses are detected. Figure 19 and 20 show different and unexpected distribution. We can see when there is no packet loss the packet distribution shows more burst than when there are packet losses. From the result, we can observe that bursty packets are strongly related to packet loss but bursty bytes (link bandwidth) are not an important factor for packet loss in underutilized links.

Table 9 respectively shows the statistics for incoming bursty bytes. From the table we can observe that the mean value of incoming bursty bytes without packet loss and with packet loss is similar for both time (Mean value of incoming bursty packets without packet loss is little higher.) scales. On the other hand, standard deviation even shows that the distribution of bursty bytes gets narrower when there is the packet loss.

Packet Loss?	Min.	Max.	Mean	Standard Dev.	Standard Dev. divided by Avg.
No	60	13948	1730	1932	1.11
Yes	0	12212	680	1187	1.74

Table 9. Incoming Burst Byte Statistics

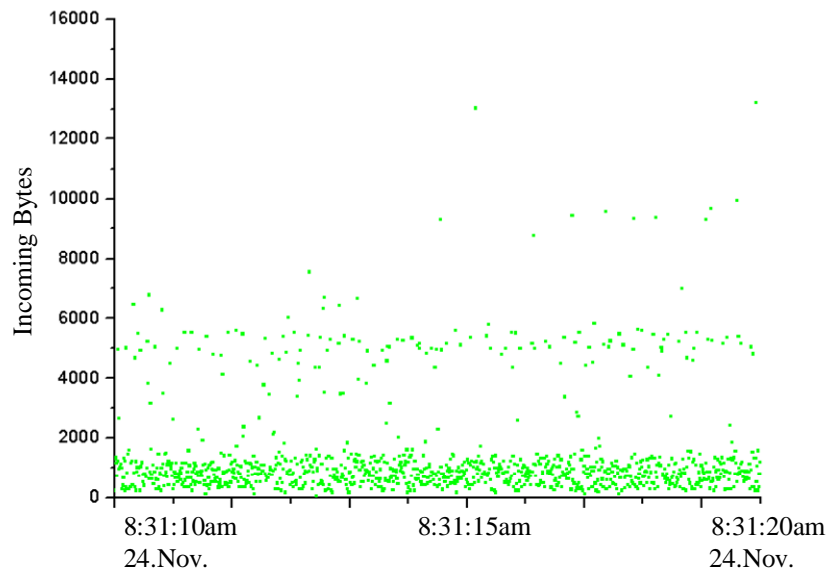


Figure 19. Incoming Burst Byte Distribution without Packet Loss

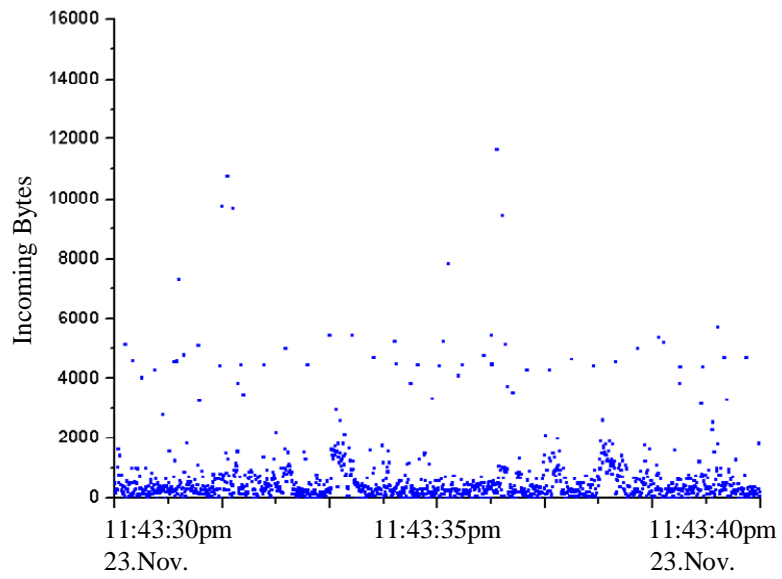


Figure 20. Incoming Burst Byte Distribution with Packet Losses

Figure 21 illustrates the interval between passing packets over ten seconds, during this period of time there were packet losses. In the graph, one dot represents one packet and each packet has its own unique number in increasing order one by one, and if there is a low degree slope, it means the interval between passing packets are small. The graph in figure 21 shows not a straight line, it means interarrival times of packets are not evenly distributed during this period. Figure 22 also illustrates the interarrival times of packets but during this period of time there were no packet loss. The graph in figure 22 shows a straight line, it means interarrival times of packets are evenly distributed during this period of time and this proves the interval between passing packets affecting packet loss.

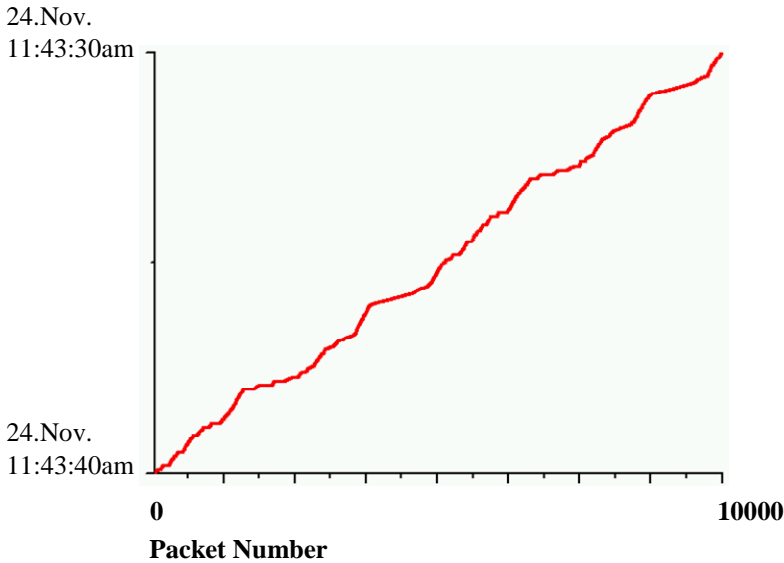


Figure 21. Interval Time between passing Packets with Packet Losses

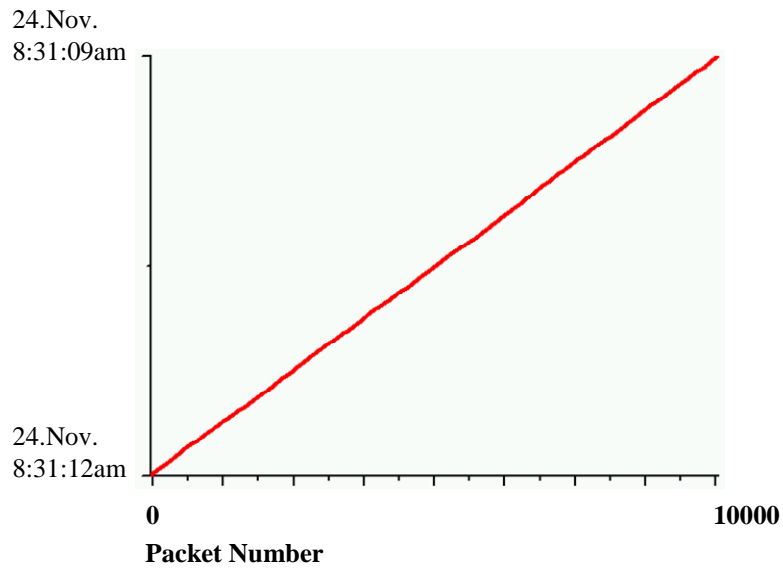


Figure 22. Interval Time between passing Packets without Packet Loss

5.3. Packet Size Distribution

The traffic monitoring system collected IP packet headers with timestamp. Figure 23 illustrates packet size distribution over one minute. In the graph, one dot represents one packet and during this period, packet loss occurred in marked time. In this graph, no special characteristics are found during the packet loss occurred time.

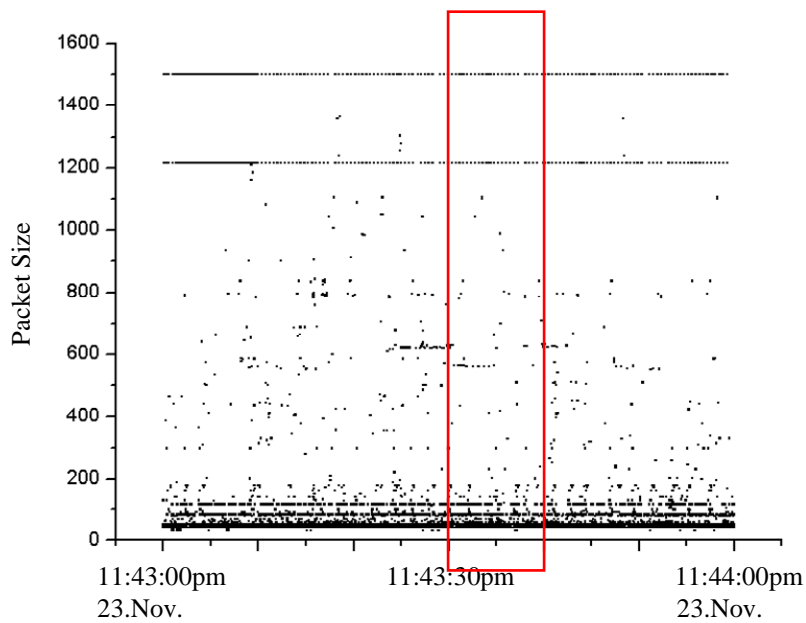


Figure 23. Packet Size Distribution

Figure 24 and 25 are zoomed in graphs of figure 23. They also illustrate packet size distribution but over 10 second. We can see no special traffic properties between graphs in figure 24 and 25. Internet traffic is generated by many different applications and certain packet sizes are more popular than others but it is not a property for packet loss. There are no traffic characteristics for packet loss related to packet size distribution on underutilized link.

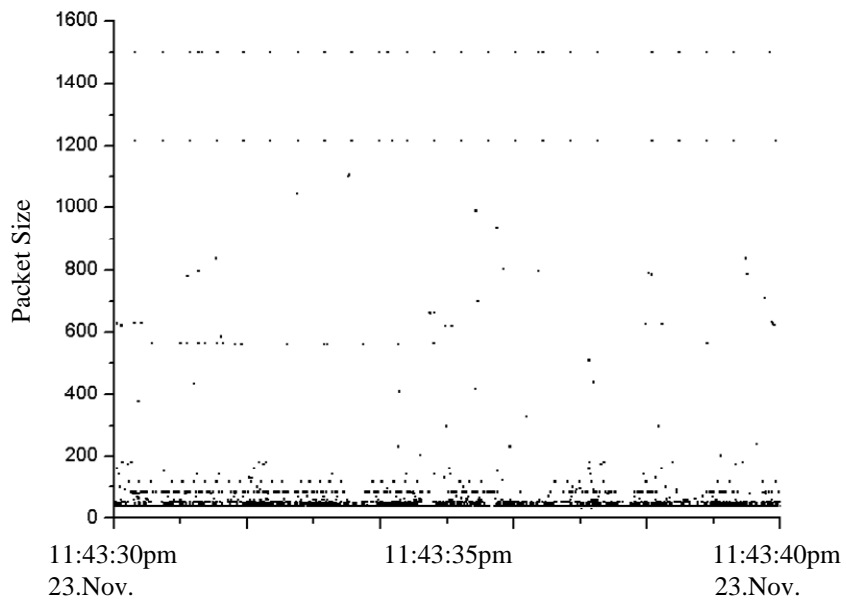


Figure 24. Packet Size Distribution with Packet Losses

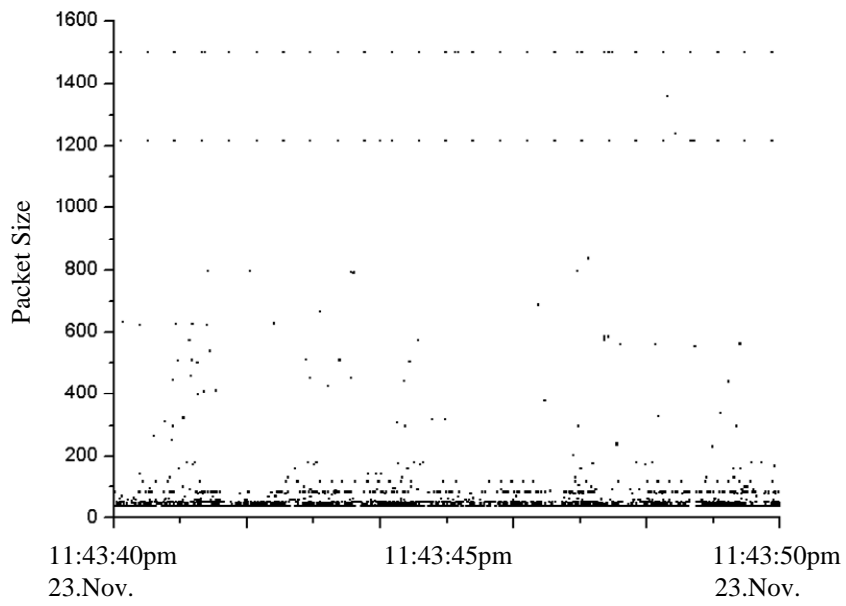


Figure 25. Packet Size Distribution without Packet Loss

5.4. Flow Analysis

The traffic monitoring system collected the number of flow (five tuple based: IP Src/Dst Address, Port Src/Dst and Protocol) data on monitored link with one second time granularity. Figure 26 illustrate the distribution of flow and the packet loss distribution. From the graph, we can see packet losses occur anytime whether the amount of flow was high or low. We can observe that packet loss is not affected by the amount of flow.

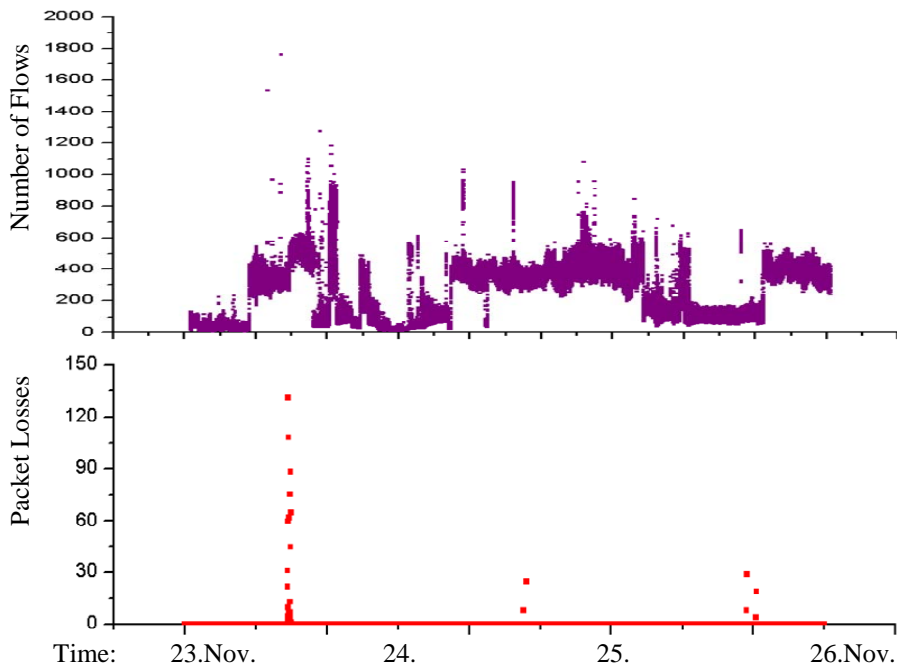


Figure 26. Flow Distribution and Packet Loss Distribution

5.4.1. Flow Life Time

We analyzed flow data into two parts: Long-life flow & Short-life flow. Long-life flow means the flow that is alive (to be alive, there should be at least one same flow occur in one minute) longer than or equal to one minute, and short-life flow means the flow that is alive shorter than one minute. Long-life flows have the high probability that the flow is generated by long time of file transformation. TCP uses the slow start algorithm. Because of TCP properties, the hub directly connected to the computers that generate long-life TCP flows occur many and continuous packet losses.

Figure 27 illustrates TCP flow distribution over 40 seconds period at 1 second measurement granularity. Packet loss occurred in marked time. As shown in the graph, there are no special characteristics when there is the packet loss. This result is true, because we monitored the link that is connected to dormitory backbone switch with 1 Gbps. Before traffic generated by personal computers reaches the dormitory backbone switch, the traffic needs to pass through several hubs and 100 Mbps links, and the effect of TCP properties (e.g., TCP slow start) that affect packet losses are almost none on the dormitory backbone switch.

Figure 28 illustrates UDP flow distribution over 40 seconds period at 1 second measurement granularity, and the graph shows same conclusion, that there is no special characteristics when there is the packet loss, as shown in the TCP flow distribution graph.

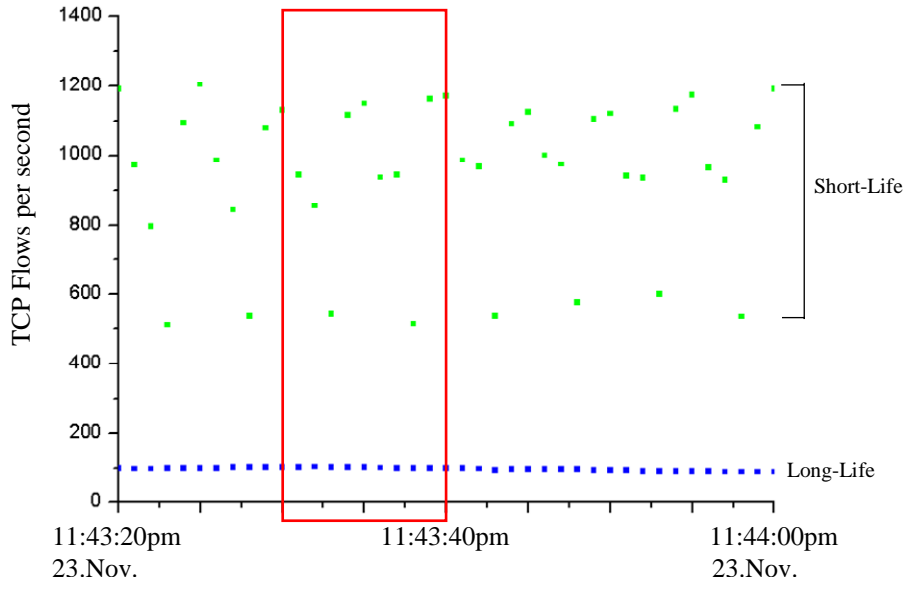


Figure 27. TCP Flow Distribution

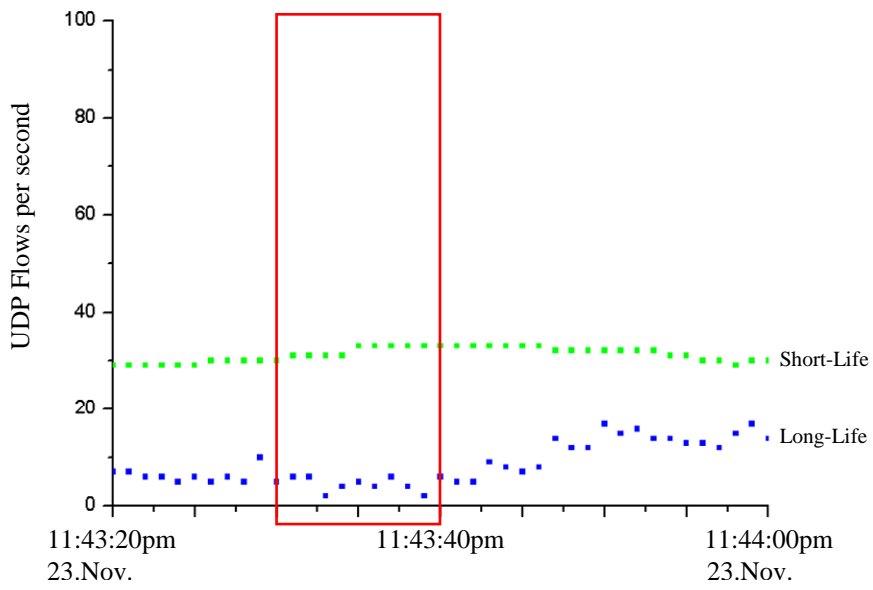


Figure 28. UDP Flow Distribution

5.4.2. One-Tuple Based Flow

Figure 29 illustrates the distribution of flow that is one-tuple based (Destination IP address) over the one minute period at 1 second measurement granularity and figure 30 illustrate the distribution of one-tuple based flow over the one minute period at 10 millisecond measurement granularity. From the previous analysis, we found the five-tuple based flow (IP Src/Dst Address, Port Src/Dst and Protocol) does not show any special characteristics related to the packet loss, so we tried to merge packets into one-tuple based (Destination IP address) flow. Because the function of the switch is switching packets depend on the packet destination IP address. The dormitory backbone switch, we monitored, does not look into the information such as Source IP Address, Source Port, Destination Port and Protocol. Figure 29 and 30 show the effect of one-tuple based flow on the packet loss.

The marked time on the graphs, packet losses are occurred. The graphs show no special flow properties on packet loss time, same as five-tuple based flow characteristics on the packet loss. We can observe the packet destination does not affect the packet loss on underutilized link.

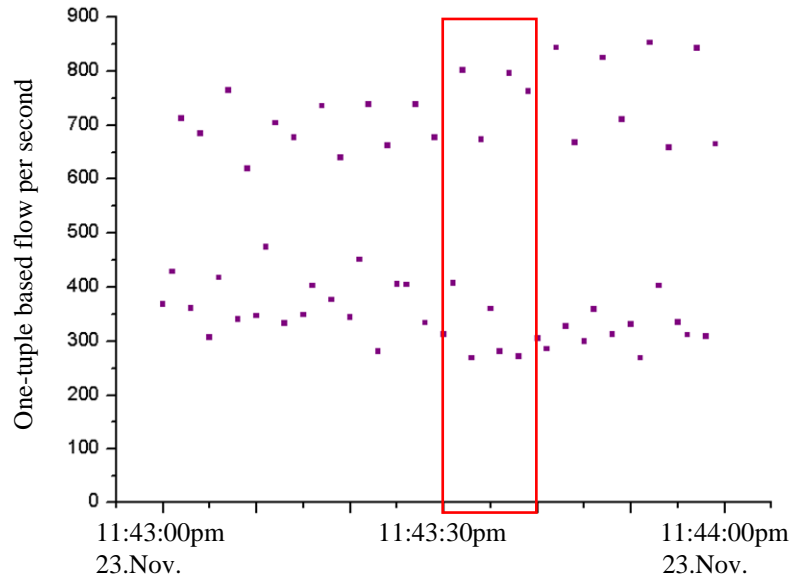


Figure 29. One-tuple based flow at 1 second granularity

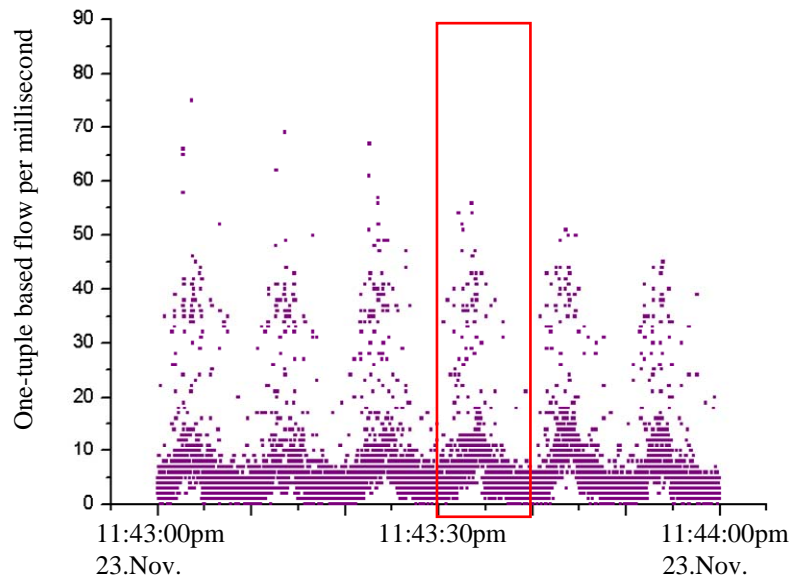


Figure 30. One-tuple based flow at 10 millisecond granularity

6. Concluding Remarks & Future Work

In our work, we collected data from the link using TAP and SNMP MIB variables from POSTECH campus dormitory network and observed IP traffic and packet losses. Analysis of this data shows that traffic bursts occur at small time granularity such as 10 seconds and 10 milliseconds, and we could find the packet losses in underutilized links.

Analysis reveals that packet losses on underutilized link are caused by a number of bursty packets more than bursty bytes in a small time scale and time interval between passing packets is an important factor for packet loss. We also observed that packet size distribution is not affecting packet losses on underutilized links. We analyzed various types of flow such as long or short-life time of TCP/UDP flow and one tuple-based flow, and found the number of flows we analyzed during this study does not affect packet loss at all. Internet traffic that is generated and merged from many applications does not show the special characteristics of the packet size and the flow on the packet loss. Only the bursty packets are affecting the packet loss on underutilized link.

In this paper, we proved the existence of packet losses on underutilized links. We monitored the link and switch using our traffic monitor modules implemented linux system and analyzed the packet loss with various types of traffic property but the 10 milliseconds time granularity can be still large time scale to find out the packet loss characteristics for router/switch process.

For future work, we are trying to monitor the packet loss in microsecond unite with the help of hardware (e.g. DAG Card) and detecting bursty CPU load at small time scale (Cisco enterprise MIB offers 5 second avg. value of CPU load.) to find more accurate relationship between CPU load and the packet loss. We are also trying to study packet loss characteristics according to different applications such as Web, FTP, and P2P on first-contact hub. We expect the network to reveal interesting properties.

References

- [1] Cisco, “NetFlow Services and Applications,” Cisco White Papers, http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neflct/tech/napps_wp.htm.
- [2] MRTG, “Multi Router Traffic Grapher,” <http://people.ee.ethz.ch/~oetiker/webtools/mrtg/>.
- [3] Se-Hee Han, Myung-Sup Kim, Hong-Taek Ju and James W. Hong, "The Architecture of NG-MON: A Passive Network Monitoring System," Distributed Systems: Operations and Management, Montreal Canada, October 2002, pp. 16-27.
- [4] Seung-Hwa Chung, Deepali Agrawal, Myung-Sup Kim, James W. Hong, and Kihong Park, "Analysis of Bursty Packet Loss Characteristics on Underutilized Links Using SNMP", 2004 E2EMON, San Diego, California, USA, October 3, 2004, pp. 68-74.
- [5] K. Park and W. Willinger. “Self-Similar Network Traffic and Performance Evaluation,” Wiley-Interscience, 2000.
- [6] Konstantina Papagiannaki, Rene Cruz and Christophe Diot, “Network Performance Monitoring at Small Time Scales,” Internet Measurement Conference, Miami, Florida USA, October 2003.
- [7] James Hall, Ian Pratt, Ian Leslie and Andrew Moore, “The Effect of Early Packet Loss on Web Page Download Times,” Passive and Active Measurement Workshop, La Jolla, California USA, April 2003.

- [8] Klaus Mochalski, Jörg Micheel and Stephen Donnelly, "Packet Delay and Loss at the Auckland Internet Access Path," Passive and Active Measurement Workshop, Fort Collins, Colorado USA, March 2002.
- [9] Siegfried Lifer, "Using Flows for Analysis and Measurement of Internet Traffic," Diploma Thesis, Institute of Comm. Networks and Computer Engineering, University of Stuttgart, 1997.
- [10] J. Case, M. Fedor, M. Schoffstall and J. Davin, "A Simple Network Management Protocol," RFC 1157, May 1990.
- [11] Cisco, "MIB Compilers and Loading MIBs," Cisco Technical Notes, http://www.cisco.com/en/US/tech/tk648/tk362/technologies_tech_note09186a00800b4cee.shtml.
- [12] Cisco, "Input Queue Overflow on an Interface," Cisco Technical Notes, http://www.cisco.com/en/US/products/hw/modules/ps2643/products_tech_note09186a0080094a8c.shtml.
- [13] LBNL's Network Research Group, <http://ee.lbl.gov/>.
- [14] W. Richard Stevens. "UNIX Network Programming, Networking APIs: Sockets and XTI," Prentice Hall, 1998.

이 력 서

성 명 : 정 승 화

생 년 월 일 : 1979 년 8 월 28 일

출 생 지 : 대한민국 서울특별시

주 소 : 경북 포항시 남구 효자동 포항공과대학교 정보통신연구소 453 호

학 력

1998.10 – 2002.02

Warsaw University of Technology, Computer Science, 졸업 (학사)

2003.03 – 2005.02

포항공과대학교, 전자컴퓨터공학부 (컴퓨터공학), 졸업 (석사)

경 력

1998.12 – 2002.03

Korean School of Warsaw

- Middle school 1st, 3rd Grade Teacher (Teaching Mathematics)

2001.11 – 2002.02

DDC Ltd.

- Network Administrator (Monitor & Control Servers)

학 술 활 동

◆ Conference Papers

- Hun-Jeong Kang, Seung-Hwa Chung, Seong-Cheol Hong, Myung-Sup Kim and James W. Hong, "Towards Flow-based Abnormal Network Traffic Detection", 2003 Asia-Pacific Network Operations and Management Symposium (APNOMS 2003), Fukuoka, Japan, October 1-3, 2003, pp. 369-380.
- Myung-Sup Kim, Hun-Jeong Kang, Seong-Cheol Hong, Seung-Hwa Chung, James W. Hong, "A Flow-based Method for Abnormal Network Traffic Detection", Proc. of the IEEE/IFIP Network Operations and Management Symposium (NOMS 2004), Seoul, Korea, April 2004, pp. 599-612.
- Seung-Hwa Chung, Deepali Agrawal, Myung-Sup Kim, James W. Hong, and Kihong Park, "Analysis of Bursty Packet Loss Characteristics on Underutilized Links Using SNMP", 2004 E2EMON, San Diego, California, USA, October 3, 2004.

연 구 활 동

◆ Projects

- DDoS 공격 완화를 위한 라우터용 트래픽 측정과 조절 기능 개발
(액세스 네트워크용 보안 라우터 시스템의 DDoS 공격 완화 기능 개발을 위한 요소 기술이 될 트래픽 측정 기능과 트래픽 조절 기능을 시험할 수 있는 Linux 기반의 시험 시스템을 개발)

감 사 의 글

2년 전 포항공과대학교에 들어 왔던 것이 엇그제 같은데, 벌써 졸업을 하게 되었습니다. 항상 시간은 빠르게 흐른다고 알고 있었지만 이렇게 다시 한번 경험하게 되니 정말 시간이 귀중함을 새삼 느끼게 됩니다. 저에게 이 자리를 예비해 주셨고, 준비해 주신 성령님의 인도하심에 감사 드립니다.

대학원이란 곳에서 2년 동안 새로운 경험을 하면서 많이 부족했던 저를 이끌어 주셨던 홍원기 지도교수님께 감사 드립니다. 겉으로는 표현을 잘 안 하셨지만, 마음으로는 늘 저가 생각지도 못한 부분들까지도 챙겨주시던 우리 교수님, 그리고 이 논문을 완성하기까지 많은 도움을 주셨던 Purdue대학의 박기홍 교수님과 주홍택 선배님께 정말 감사하다는 말을 전하고 싶습니다. 연구실 랩장이시던 김명섭 선배님, 교수님께 무엇인가를 제출 하기 전에 항상 먼저 보시고 수정해 주시던 기억이 납니다. 선배님을 통해 정말 많은 것들을 배울 수 있었습니다. 또한 강훈정 선배님, 처음 연구실에 들어와서 선배님과 함께 논문을 쓰던 기억이 납니다. 코드를 작성하는데 있어서 많은 도움을 주셨었는데, 감사합니다. 그 외에도 최미정 선배님, 최현미 선배님, 김은희 누나, 최영미 누나, 김동현 형, 조룡권 형, 홍성철, 이소정, 원영준, 이형조, 유선미 그리고 Deepali 연구실에 있으면서 많은 시간을 함께 했던 우리 연구실 원들 모두에게 감사하다는 말을 하고 싶습니다.

폴란드에 계시는 저의 부모님과 동생 신형, 제가 이곳에서 이렇게 지낼 수 있었던 것은 저의 가족의 기도 덕분인 것 같습니다. 항상 저를 생각하시며 늘 걱정해 주시고 위로해 주시던 저희 가족에게 감사를 전하고 싶습니다. 그리고 사랑합니다.

포항이라는 곳은 저에게 지식만이 아니라 영적으로도 큰 변화를 주었습니다. 공대의 영성을 위하여 그리고 공대교회를 위하여 늘 기도하시는 정진욱 교수님과 신우회 교수님 분들 그리고 하정애 전도사님, 학생으로 있으면서 바쁜 시간 속에 공대를 위해 그리고 저를 위해 기도해 주신 성상학 형제님과 김수진 자매님, 이진욱 형제님과 신진향 자매님, 김한나 자매님, 윤준호 형제님, 박진석 형제님, 그리고 김지태 형제님 이 외에도 많은 형제, 자매님들이 있었습니다. 정말 감사합니다.

이 외에도 저를 위해서 기도해 주시고 격려해 주던 저의 오랜 친구들과 인터콥 선교회분들, 세계로 선교회분들과 효자교회 분들 모두에게 감사의 마음을 전하고 싶습니다.