

Characteristic analysis of internet traffic from the perspective of flows

Myung-Sup Kim*, Young J. Won, James W. Hong

Department of Computer Science and Engineering, POSTECH, San-31, Hoyja-dong, Nam-gu, Pohang, Kyungbuk 790-784, South Korea

Available online 26 August 2005

Abstract

The necessity of network traffic monitoring and analysis is growing dramatically with increasing network usage demands from individual users as well as business communities. Most network traffic monitoring and analysis systems are based on flows. One key asset with these systems is to compress a significant amount of packet data into flows. However, the compression ratio is highly variable in the recent network environments due to the increased use of peer-to-peer file sharing applications and the frequent appearances of abnormal traffic caused by Internet worms, which negatively influences the performance of traffic analysis systems. The performance of traffic monitoring and analysis systems highly depends on the number of flows as well as link utilization and the pattern of packet arrival. This paper examines the characteristics of recent Internet traffic from the perspective of flows. We found that the frequent occurrence of flash flows highly affects the performance of the existing flow-based traffic monitoring systems. Using various flow-related metrics, we analyzed the IP traffic traces collected from the Internet junction at POSTECH, a university with over 6000 end hosts and servers.

© 2005 Elsevier B.V. All rights reserved.

Keywords: Internet traffic; Passive monitoring; Traffic analysis; Traffic flow; Flash flows

1. Introduction

The necessity of network traffic monitoring and analysis is growing dramatically with the increasing network usage demands from individual users as well as business communities. Traditional areas, which are heavily dependent on traffic monitoring, cover a range from network capacity planning to the study of network behavior. In addition, emerging new areas, such as service level agreement (SLA), customer relationship management (CRM), quality of service (QoS), security attack analysis, and usage-based billing, also have great needs for traffic monitoring.

Most traffic monitoring and analysis systems [1–3] focus on flow-based investigation. The definition of a traffic flow

may be different depending on systems [4–6]. The most widely used definition of a traffic flow is the one from Cisco Systems, which announced several versions of flow definitions with its NetFlow series. Among them, NetFlow V5 [6] is most commonly used in real world applications. NetFlow V5 defines a traffic flow as a unidirectional sequence of packets between given source and destination endpoints. Traffic flows are highly granular; flow endpoints are identified both by IP address as well as by transport layer application port numbers. NetFlow also utilizes the IP protocol type, type of service (ToS) and the input interface identifier, to uniquely identify flows. In addition, the IETF working group IPFIX [7] is trying to standardize an IP flow format.

The process of flow-based traffic monitoring starts from capturing and classification of packets according to flow identifiers from packet header values and generates flow data. Some systems, such as NG-MON [1] and Ntop [2], capture raw packets directly from a network link or a network device and generate flow data using their own flow format. Cisco IOS-based routers and switches are equipped with a function to export flow data in the NetFlow format [6].

One of the key assets of these systems is to compress a significant amount of packet data into flows. However, the compression ratio from packet header data to flow data

* Corresponding author. Address: Network Architecture Laboratory, Department of Electrical and Computer Engineering, University of Toronto, Rm4187, 40 St George Street, Ont., M5S1A4, Canada. Tel.: +1 416 946 7059; fax: +1 416 978 8676.

E-mail addresses: mount@postech.ac.kr, myungsup.kim@utoronto.ca (M.-S. Kim), yjwon@postech.ac.kr (Y.J. Won), jwkhong@postech.ac.kr (J.W. Hong).

is highly variable in the recent IP network environment due to the increasing use of peer-to-peer (P2P) file sharing applications [8] and frequent appearances of abnormal traffic (e.g. scanning, DoS/DDoS, Internet worms) [9], which negatively influences the performance of traffic analysis systems. Abnormal traffic can be problematic because it usually generates a large number of flows, which consist of only a couple of packets per flow. The system performance highly depends on the number of distinct flows as well as the link utilization or the number of packets. Therefore, high performance monitoring systems are desirable even for low-speed/under-utilized network links.

This paper studies the characteristics of recent Internet traffic from the perspective of flows and investigates the cause for the high fluctuation in the number of flows. It also presents our proposition to enhance existing flow-based traffic monitoring systems. We analyze various flow measurement metrics, which we defined in this work, with traffic traces collected on the Internet junction at POST-ECH, a university with over 6000 end hosts and servers.

The organization of this paper is as follows. Section 2 describes the influence of flows on the traffic monitoring systems and recent research on Internet traffic analysis. Section 3 describes our traffic data collection method. The high-level analysis of IP traffic trace is given in Section 4. In Section 5, we describe the detailed analysis of IP traffic flows. Findings from our investigation and some suggestions for a traffic analysis system are given in Section 6. Finally, concluding remarks are given and possible future work is discussed in Section 7.

2. Related work

In this section, we describe the system architecture widely accepted by real-time traffic monitoring systems and explain the influences of traffic flows on their system performance. We also summarize the trends and results of current Internet traffic analysis by various research groups.

2.1. Influence of traffic flows on traffic monitoring systems

The architecture of real-time traffic monitoring and analysis systems are evolving from single system architecture to pipelined and clustered architecture as target link speed is growing from several hundreds Mbps to multi-Gbps. The original version of Ntop [2] and argus [10] are good examples of the single system architecture, which may still be deployed on under-utilized and low-speed network links. However, most real-time monitoring systems use pipeline-based architecture, which is basically influenced by the IETF RTFM architecture [11] to cope with high-speed network monitoring needs. Systems such as NetTraMet [11], Ntop [2], NetFlow-based systems [3], NG-MON [1] and Ahang Shiyong et al. [12] introduced the clustering concept to pipeline-based architecture to handle a large

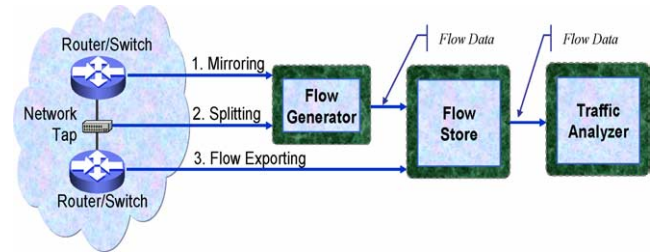


Fig. 1. Pipeline-based real-time traffic monitoring and analysis system architecture.

amount of traffic data in a real-time manner for use with commodity off-the-shelf hardware. Also, flow-based systems provide a scalable and flexible way to interpret various analysis requirements (e.g. accounting/billing, SLA, QoS monitoring, and etc.). System architects introduced the idea of flow to the pipelined architecture in order to minimize the impact on performance and system overhead.

The flow-based traffic monitoring system architecture commonly consists of three major processing parts, as illustrated in Fig. 1: flow generator, flow store, and traffic analyzer. A flow generator is located at the measurement point to capture packets using mirroring or splitting functions and export flow record data. It can be by-passed by utilizing flow exporting functions embedded in network devices, such as Cisco NetFlow exporting. A flow store is the collection and storage point for data before it is passed to a traffic analyzer. There can be multiple flow generators or flow stores wherever they are necessary. Considering the effects of many traffic metrics (such as link utilization, packet arrival rate, and the number of flows) on the traffic monitoring system, there are some performance issues in each phase of the pipelined monitoring system architecture.

First, the performance of flow generator appears to be highly affected by link utilization and packet arrival rate rather than the number of flows, because the packet capturing function consumes more system resources than the flow generating function. However, the number of flows also significantly affects the performance of a flow generator, specifically the required memory size and search processing load. For example, if a flow generator uses a static hash bucket size, the system may be corrupted when the number of flows exceeds the limit.

With the help of SmartBits™ 600, a packet generation hardware, we tested the effect of the number of flows generated on the performance of flow exporting systems. For this test we used nProbe [2] and NG-MON's flow generator [1]. nProbe can generate Cisco's NetFlow flows and mimic the router's flow exporting routine. It allows the users to change a hash bucket size for various link conditions; the hash bucket size is static at run-time. On the other hand, the NG-MON flow generator uses a variable hash bucket size using the open hash technique.

While fixing the packet arrival ratio (50,000 pps), we changed the number of distinct flows from 10 to 1,000,000 sending from the SmartBitsTM. nProbe could not export any more flows after its hash buffer was full and it took up the entire CPU load until its termination. In case of the NG-MON flow generator, as the number of distinct flows increased we monitored a slight increase in CPU load and a much more noticeable growth in memory consumption. From this test, we confirmed that the number of flows highly affects the required memory size in the flow generation with a little effect on CPU load. Experimental research is on-going to overcome this high memory consumption such as adaptive flow exporting scheme and developing a new flow format with an effective sampling mechanism [4,13,14].

Second, the data exchange overhead and delay increases as the flow data volume increases, not the link utilization or packet size. This is obvious because a single-flow record size is fixed regardless of the number of packets and byte count of the flow. The large amount of transferred flow data may hinder other service traffic if a monitoring system is running on the operational network. Some ISPs and companies find a way around these problems by constructing a completely separate network just for monitoring and control purposes. Although such network consists of low bandwidth links and cheaper network devices, it is still a very costly solution.

Third, the limited resources for the analysis application cause system overhead handling a large number of flows from today's traffic. To meet the requirements for sophisticated analysis reports, many systems need to provide support for various and complicated traffic metrics. Because we are dealing with large amounts of data, frequent disk operations are necessary which obviously slows down the analysis process. For example, we sent 100,000 distinct packets at 20,000 packets per second (pps) to the system where Ntop was running and witnessed the CPU load reaching 99%. 100,000 distinct packets mean that Ntop must process 100,000 flows to show the analysis data of flow measurements. Although the figures obtained in this test can change when faster hardware is used, the analysis system performance will remain highly dependent on the number of distinct flows.

2.2. Current status on Internet traffic monitoring and analysis

The types and patterns of current network traffic are more complex than in the past. The proportion of traditional well-known port based traffic is decreasing. P2P, streaming media, and game traffic are growing instead. Internet2 administrator report that about 16% of the traffic carried by their network is P2P traffic while a further 54% of unidentified traffic may also belong to P2P applications [15]. Many other ISPs also believe that their networks are overwhelmed with P2P application traffic [8,16].

A number of studies involving current Internet traffic analysis concentrate on the characteristic analysis of the newly emerging application traffic. The preliminary step of traffic characterization is to classify the traffic according to application layer programs. However, it becomes increasingly difficult to detect newly emerging Internet applications which use dynamically generated port numbers rather than static and registered ones. Here are a few different approaches to this traffic identification task.

3. Traditional identification method

The traditional method is based on the well-known ports registered at the IANA port list or the private port list extended from IANA port list [17]. For example, web traffic created by web client/server applications use port numbers 80, 8080, or 443. However, we cannot rely on this method any more because of many new features in recent Internet traffic. For example, traditional methods cannot detect the dynamic port numbers negotiated by streaming media applications, such as Microsoft window media server/player.

4. Payload examination based method

To detect the dynamically determined ports of streaming media applications and P2P applications, the payload examination method is one possibility. Mmdump [18] and SM-MON [19] used this method to differentiate streaming media traffic from other Internet traffic. They examine the payload of control packets to determine the port numbers used by data session. In addition, it is also useful in detecting passive FTP data sessions.

5. Signature mapping based method

To increase identification accuracy, the signature mapping based method [20] was introduced. In this method, a portion of payload data that is static, unique, and distinguishable from other packets is investigated for all applications. It can be recorded as the signature of that application. By comparing every packet payload with a pool of pre-determined signatures, this method can identify application traffic more accurately than the traditional method.

Recent studies on traffic characterization collected and analyzed IP traffic from an enterprise network [21] or a large ISP backbone network [22] while focusing on traffic characteristics from various perspectives: User perspective characteristics, packet-level and flow-level features [22], and routing protocol-level behaviors [23]. Numerous studies [24–26] contributed methods to ascertain the nature of P2P traffic. They particularly looked at the traffic generated by

FastTrack (KaZaA, KaZaA Lite), Gnutella (Morpheus, LimeWire), and Overnet (eDonkey), which consume a significant share of Internet traffic. Other research [8,16] focused on the comparison of P2P traffic with other traditional Internet traffic, such as Web and FTP.

Sen et al. [24] investigated TCP flow-level data gathered from multiple routers across a large Tier-1 ISP to analyze three P2P applications (KaZaA, Gnutella and DirectConnect). This research was a significant step towards characterizing these P2P applications from a network engineering perspective. However, their work did not reveal application level details and insights explaining the observed behaviors.

Two studies [8,27] considered that KaZaA P2P traffic could be logged and cached. Although KaZaA's control protocol is proprietary, these studies focus on KaZaA using HTTP for transferring data files. They monitored HTTP traffic on highly utilized links: Traffic from a large Israeli ISP to US and Europe [27], and traffic from the University of Washington campus to ISP [8]. Both reported that most Internet traffic constitutes KaZaA traffic. Furthermore, they compared KaZaA traffic with traffic generated by traditional content distribution systems, such as Akamai and Web traffic [8]. They quantified the rapidly increasing P2P traffic, characterized the behavior of these systems from the perspectives of clients, objects, and servers, driving the appropriateness for caching.

5.1. Collection of IP traffic trace

In order to collect IP traffic trace data, we deployed a NG-MON flow generator on the Internet junction at POSTECH, as illustrated in Fig. 2. NG-MON is a real-time traffic monitoring and analysis system for high-speed network links, which was developed by our research group in 2002 [1]. The POSTECH campus Internet link is composed of two 100 Mbps Metro Ethernet links. There are two core switches and two Internet routers connected with four 1 Gbps Ethernet links in a mesh structure. We used four optical taps to collect all in/out Internet traffic from the four 1 Gbps Ethernet links between core switches and Internet routers.

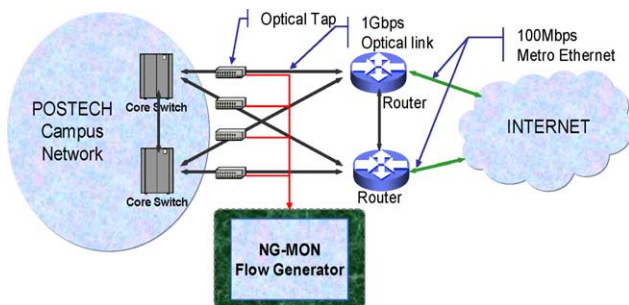


Fig. 2. Traffic trace collection method.

We have about 6000 computers connected to the POSTECH campus network. Microsoft Windows Operating Systems (Windows 98/ME/XP/NT/2000) are installed in 80% of computers, and the remaining 20% are running on Unix-like operating systems, such as Linux, Sun Solaris, IBM AIX, HP-UX, etc. In fact, POSTECH's 2800 students all live in campus dormitories.

The NG-MON Flow Generator receives raw packets and generates flow data. For our purposes, we define flow as a unidirectional stream of packets with the same five-tuple packet header values; source IP, destination IP, source port, destination port, and protocol number. In addition, we used the flow timeout value of 120 s based on the research results of Gianluca Iannaccone et al. [28]. When two consecutive packets with same five-tuple values have more than 120 s of interval we consider these two packets belong to two different flows. In this experiment, first we gathered traffic data during three different weeks and generated flow data based on our flow definition. After generating flows, we excluded the flows started before our testing period and continued after the testing period.

We collected IP traffic for three weeks during February and March 2004. The overall summary of our traffic trace is illustrated in Table 1. The total number of flows captured during three weeks was 2610×10^6 , and the total bytes were over 40 TB. Among them, we considered only TCP and UDP traffic, which occupies more than 99% of total traffic in bytes. From the captured traffic data, we excluded the packets, which had spoofed information in the packet header values (e.g. an outgoing packet with source IP address which does not belong to the IP range of our campus network, a packet with fabricated TCP flags, and etc.). The portion in this category was 0.5% in bytes and 3.3% in packets of total traffic.

The average bytes per packet were calculated as 642 b from Table 1. The average bytes per TCP packet was 678 b, which was greater than that of UDP traffic (239 b). The average packet count of flow was 28 (average TCP and UDP packet counts of flow were 98 and 3, respectively). We discovered that a large number of UDP flows are composed of less than 2 packets. The average bytes per flow were 18,239 b. Average TCP and UDP bytes per flow were 67,043 and 756 b, respectively. This result well describes the usage of TCP and UDP. TCP is used to transfer an important and large amount of data between the client and server due to its reliable service mechanism, while UDP is usually used to send short messages, the drop of which could be tolerable.

The ratio of TCP and UDP traffic in bytes and packets are similar to each other; over 90% of total packets and total bytes are TCP traffic. Still, TCP is used by the majority of current Internet applications. However, the flow ratio of TCP and UDP traffic is opposite. The number of total UDP flows is about two times greater than the number of total TCP flows, as illustrated in Table 1. A small number of UDP packets with fewer bytes than TCP packets cause

Table 1
Summary of traffic trace

Collection period	2/1/2004–2/7/2004	2/17/2004–2/23/2004	3/6/2004–3/12/2004
Total file size	41 Gb	43 Gb	53 Gb
Flows ($\times 10^6$)			
TCP	295 (34%)	13,697 (98%)	13,619 (97%)
UDP	537 (63%)	190 (2%)	327 (3%)
ICMP	33 (3%)	16 (0%)	15 (0%)
Others	0.1 (0%)	0.6 (0%)	0.2 (0%)
Total	866 (100%)	13,905 (100%)	13,962 (100%)
Bytes (Gb)			
TCP	18,345 (93%)	19,246 (93%)	21,015 (92%)
UDP	1089 (6%)	1381 (7%)	1564 (6%)
ICMP	190 (1%)	177 (0%)	335 (1%)
Others	1 (0%)	1 (0%)	2 (0%)
Total	19,636 (100%)	20,806 (100%)	29,917 (100%)
Packets ($\times 10^6$)			
TCP	295 (34%)	325 (35%)	206 (25%)
UDP	537 (63%)	543 (60%)	591 (70%)
ICMP	33 (3%)	39 (5%)	38 (5%)
Others	0.1 (0%)	0.1 (0%)	0.004 (0%)
Total	866 (100%)	908 (100%)	836 (100%)
Bytes (Gb)			
TCP	14,321 (97%)	14,321 (97%)	14,321 (97%)
UDP	341 (3%)	341 (3%)	341 (3%)
ICMP	33 (0%)	33 (0%)	33 (0%)
Others	0.5 (0%)	0.5 (0%)	0.5 (0%)
Total	14,697 (100%)	14,697 (100%)	14,697 (100%)

Table 2
Inbound traffic vs. outbound traffic

Collection period	2/1/2004–2/7/2004			
	Flows (in: out) (%)	Packets (in: out) (%)	Bytes (in: out) (%)	
Total	TCP	(47:53)	(50:50)	(41:59)
	UDP	(51:49)	(53:47)	(69:31)
	ICMP	(47:53)	(78:22)	(76:24)
	Other	(29:71)	(37:63)	(85:15)
	Total	(50:50)	(50:50)	(42:58)

a significant amount of flows. This implies that the UDP traffic in the current network environment highly influences the flow-based traffic analysis systems negatively, because the processing load of these systems depends on the number of generated flows rather than the number of packets and link utilization, as mentioned in Section 2.

Another interesting fact about flows is illustrated in Table 2. The inbound and outbound traffic show a close one to one ratio in terms of flow count and packet count. The outbound traffic means the traffic generated from inside the campus network and going out to the Internet, the inbound traffic is vice versa. However, considering bytes, the total outbound traffic is 1.38 times greater than inbound traffic, which is commonly reported in many university networks [8]. This implies that the inbound packet size is smaller than the outbound packet size, and the inbound byte size of a flow is also smaller than that of outbound flow. Moreover, the outbound TCP bytes are larger than inbound TCP bytes with the same ratio of total traffic, but the inbound UDP bytes are two times larger than outbound UDP bytes. We believe that the former case is mainly due to P2P traffic and popular FTP servers operated by students, and the latter case is due to multimedia service traffic from outside servers using UDP to transfer video/audio data.

5.2. High-level analysis on flows

In this section, we present high-level and general analysis results of the traffic trace. To characterize recent IP traffic we performed various analyses over three flow-related metrics: flow duration, packet count, and byte count. From this general analysis, we found that a large amount of flows are short-lived flows, which have small number of packet and byte counts.

5.3. Distribution of traffic over time

Fig. 3 illustrates six time-series graphs of the traffic trace captured from February 1, 2004 to February 8, 2004. The graphs show the variances of each transport layer protocol (TCP, UDP, and ICMP) traffic and their total amount in three different analysis metrics (flow, packet, and byte) over time. We also categorized the traffic into inbound and outbound traffic to compare the directional behavior of our

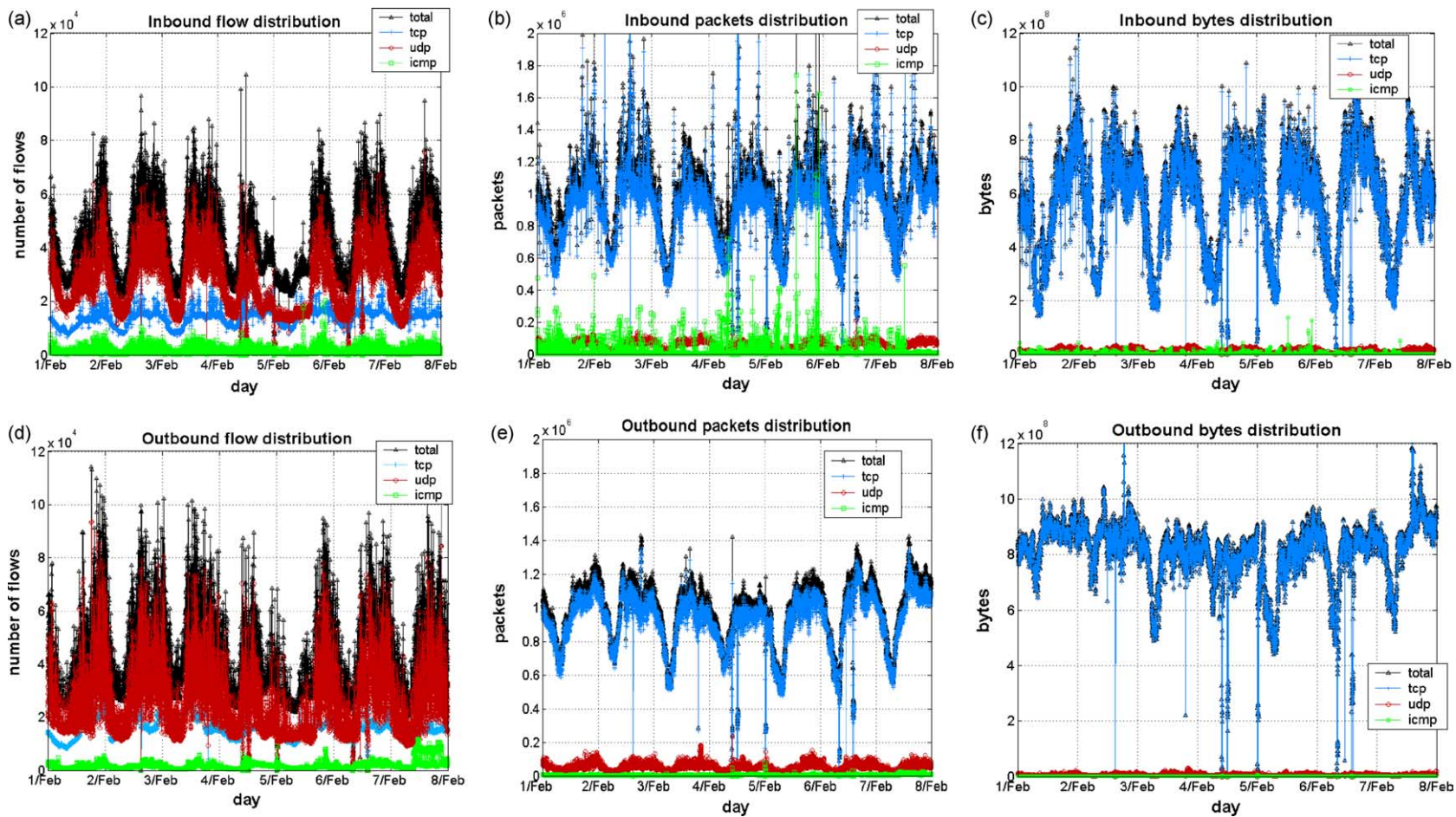


Fig. 3. Time-series graphs in three analysis metrics (flow, packet, and byte) for traffic trace captured during 1–8 February 2004.

campus network. We calculated each value with one minute of time granularity. Therefore, the graph is drawn with 10,080 different time units ($60 \text{ min} \times 24 \text{ h} \times 7 \text{ days}$) along the x -axis.

The total flow distribution is mainly affected by the UDP flows, as illustrated in Fig. 3(a) and (d). The inbound and outbound flow distribution has a similar shape and the average number of outbound flows is slightly larger than that of the inbound flow, which describes the data in Table 2. The shapes of packet distribution and byte distribution graphs are primarily affected by the amount of TCP traffic, which contradicts the shape of flow distribution.

The time-of-day feature appears in all six graphs in Fig. 3. The traffic increases from the afternoon and reaches a peak between 10 p.m. and 1 a.m. and it goes down in the morning, which is typical of our university Internet usage behavior. The incoming ICMP packets are much larger than the outgoing ICMP packets, as illustrated in Fig. 3(b) and (e). This implies that the outside IP addresses more frequently join and leave the network than the inside IP addresses. The fluctuation of incoming bytes is higher than the outgoing bytes. That is because the number of outside users is much higher than that of inside users. In other words, the more users access a network, the less fluctuation of download traffic appears.

The data from Table 1 and Fig. 3 implies some considerations about traffic flows. The flow characteristic of traffic shows different features from the byte and packet characteristics. The UDP flows are more important than TCP flows in the perspective of traffic flows. Therefore, we must consider the flow-level features of network traffic as well as the packet- and byte-level characteristics.

5.4. Distribution of flow duration

Fig. 4 illustrates the distribution of flow duration in the form of a distribution function (DF) graph and a cumulative probability density function (CPDF) graph of the traffic trace from March 6, 2004 to March 12, 2004. The total number of flows is about 852×10^6 . Among them,

the number of UDP flows is 592×10^6 , which is 2.7 times more than the number of TCP flows.

The average flow duration of TCP flows is 57.32 s, which is 5.3 times greater than that of UDP flows (10.72 s). As Fig. 4(a) illustrates, we can observe some long-lasting flows over 10^5 s (about 1 day). The maximum value of flow duration is 392,217 s, which is 4 days 12 h 56 min 57 s. The median of TCP and UDP flow duration is 1 s, which indicates that the flow duration of over 50% of total flows is less than 1 s. It is worthwhile to investigate the features of these short-term flows to improve the performance of the traffic analysis system. The standard deviations of TCP and UDP flows are 540.84 and 76.62 s, respectively. This indicates that the flow duration of most UDP flows belong to a small range of time intervals.

The number of UDP flows less than 80 s long is greater than the number of TCP flows, as illustrated in Fig. 4(a). By contrast, over 80 s long TCP flows are much more than the number of UDP flows. The number of UDP flows less than 10 s is 508,074,860, which is 3.4 times of TCP flows of the same duration. This number is 85.74% of total UDP flows, as illustrated in Fig. 4(b). Moreover, the UDP flows less than 100 and 1000 s are 97.76 and 99.97% of total UDP flows, respectively. The duration of most UDP flows is less than 1000 s. In case of TCP flows, the TCP flows less than 10, 100, and 1000 s are 68.64, 92.19, and 99.47%, respectively. From this flow duration analysis we know that the duration of TCP flows are more evenly distributed between 0 and 1000 s than UDP flows.

According to Fig. 4(c), the percentage of flows whose duration is less than or equal to 1 s is more than 60%. The ratios of such flows in total TCP flows and UDP flows are over 20 and 80%, respectively. It is an interesting fact that the overall percentage of short duration flows is high.

5.5. Distribution of packets in flows

Fig. 5 illustrates the distribution of packets in flows with the form of DF graph and CPDF graph from the same traffic trace, which is also drawn in a log scale. Fig. 5(a) shows that

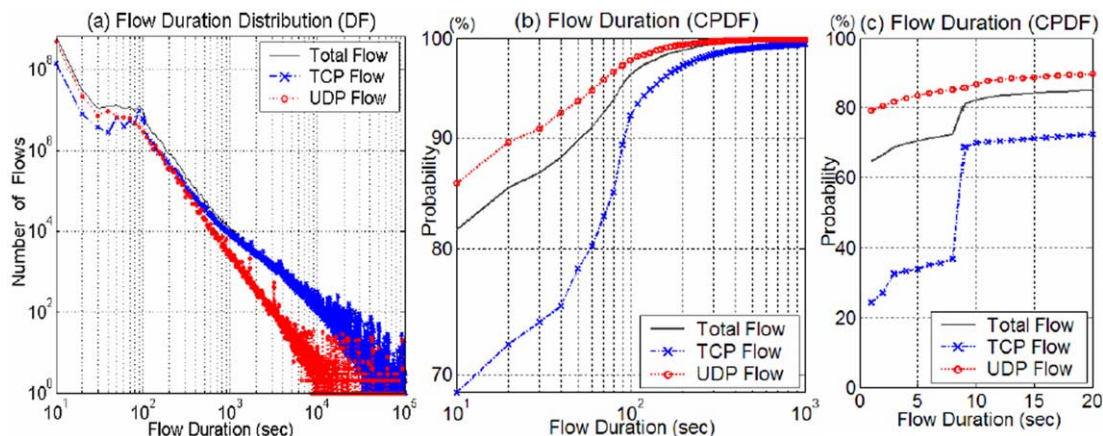


Fig. 4. Distribution of flow duration.

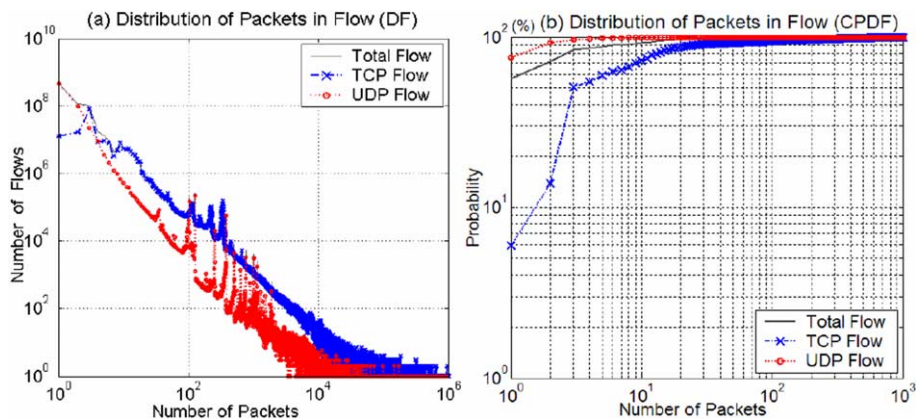


Fig. 5. Distribution of number of packets in flows.

the distribution of the total flows is similar to the distribution of TCP flows when the packet count exceeds three. The number of UDP flows with less than three packets is 19 times more than that of TCP flows. When the packet count exceeds three on the X-axis, the number of TCP flows is greater than the number of UDP flows.

Fig. 5(b) shows that the ratio of TCP flows aggregated with only one packet is about 6% of the total TCP flows, compared to about 76% in the case of UDP flows. The number of TCP flows with less than 1000 packets occupies a large portion of the total TCP flows. By contrast, the number of UDP flows with less than 10 packets takes a large portion of the total UDP flows. Particularly, the number of UDP flows with a couple of packets takes about 92% of the total UDP flows. Consequently, the number of packets belonging to the TCP flows is greater than the number of packets of UDP flows.

5.6. Distribution of bytes in flows

Fig. 6 illustrates the distribution of bytes in flows with the form of DF graph and CPDF graph from the same traffic trace, which are also drawn in a log scale.

Considering TCP flows, the byte counts of TCP flows are evenly distributed until 1000 b with some fluctuation. The number of TCP flows decreases exponentially after 1000 b. The number of UDP flows shows similar exponential decrease after 300 b. Considering the flows having less than 1000 b, the number of TCP flows is 154×10^6 , which is 72% of total TCP flows. Fig. 6(b) shows three vertical increases of TCP flows at 64, 130, and 200 b, respectively. About 90% of TCP flows are composed of less than 4000 b.

The number of UDP flows less than 1000 b is 587×10^6 , which is 99% of total UDP flows and 3.9 times larger than TCP flows. The 64 b UDP flows are 53%, which means that half of the total UDP flows are single packet flow. 90% of UDP flows are less than 200 b, as illustrated in Fig. 6(b).

5.7. Duration vs. packets vs. bytes

Fig. 7 illustrates the relationship between the three metrics (duration, packets and bytes) in flows. We also compared the TCP and UDP flows using these three values together. We used a randomly selected 800,000 flows from our traffic trace to plot these graphs. It turned out that two-thirds of these random flows were UDP flows.

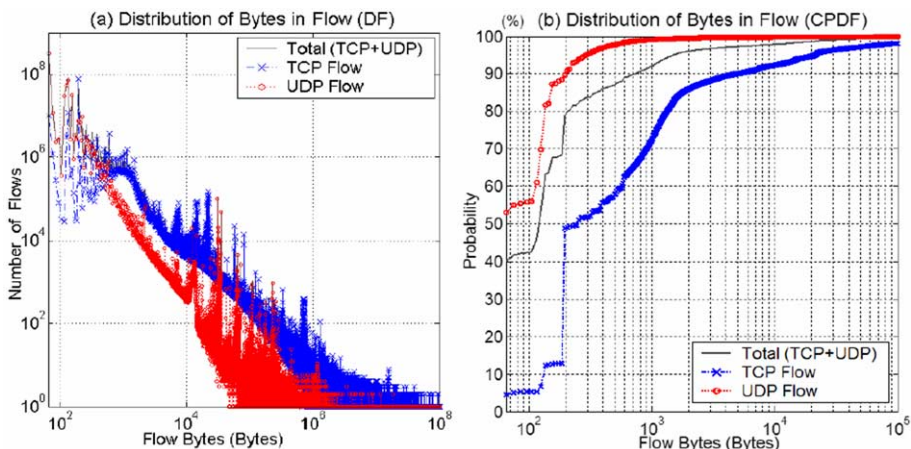


Fig. 6. Distribution of byte size in flows.

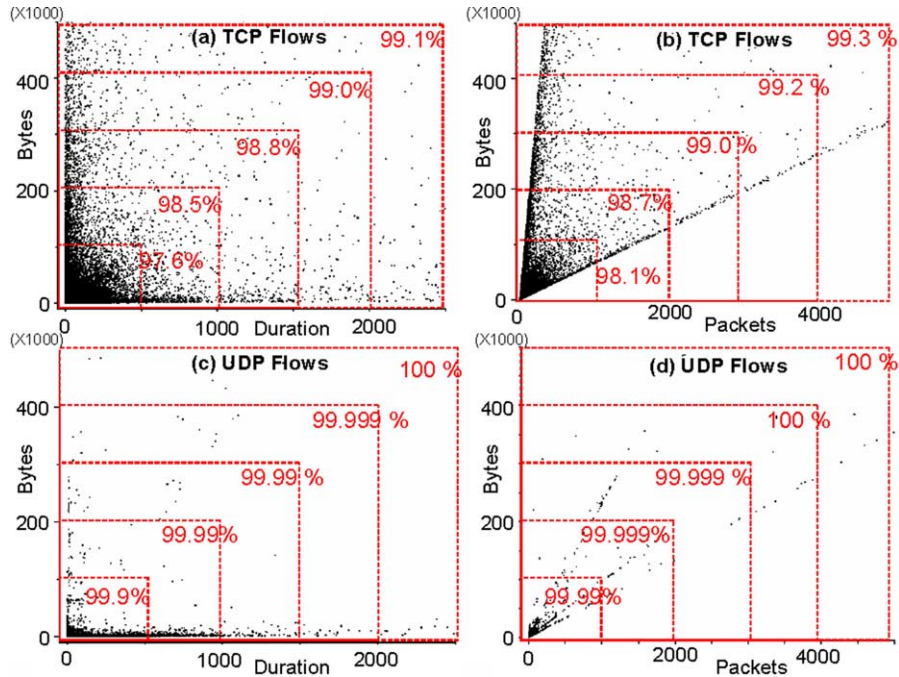


Fig. 7. Relationship among duration, packets, and bytes.

Fig. 7(a) and (c) show that the bytes in the flows are independent (but not completely) of the flow duration. The relationship between packets and flow duration is also independent of each other. They also have similar graphs to the bytes vs. duration graphs. While small-bytes and short-time flows appear in the chart, large-bytes and short-time flows appear together. The bytes in most UDP flows are less than 5000 b regardless of the flow duration. But the bytes of TCP flows are spread widely in the chart. The flows with large bytes and low duration and flows with small bytes and long duration appear together in this graph. The density of the flows in the Fig. 7(a) and (c) shows that over 97% of TCP and UDP flows are less than 100,000 b and 500 s.

The bytes in flows are proportional to the number of packets, as illustrated in Fig. 7(b) and (d). In the bytes vs. packets graph of TCP flows, two thick boundary lines appear, and all TCP flows belong between these two boundary lines. The bytes per packet near the lower

boundary are 64 b, and the bytes per packet near the upper boundary are 1518 b, which is the maximum packet size of an Ethernet frame. We had a considerable amount of TCP flows with 1518 b per packet, while little UDP flows had this amount of bytes per packet value. Most UDP flows had less than 500 packets and 50,000 b.

According to the zoomed-in duration vs. bytes distribution graph in Fig. 8, about 60% of flows lie within 1 s and 500 b. About 83% of flows also share the same characteristic in which they consist of less than three packets and again 500 b in total. Hence, a significant amount of flows have condition where they stay below 1 s long, three packets, and 500 b.

Fig. 9 illustrates the flow distributions over the three factors on a single graph: duration, packets, and bytes. More than 50% of these sampled flows share the same characteristic where duration ≤ 1 , packets ≤ 3 , and bytes ≤ 500 . In this paper, we call these short-time, small-packet-size, and small-bytes size flows as *flash flow*. For the detailed analysis on

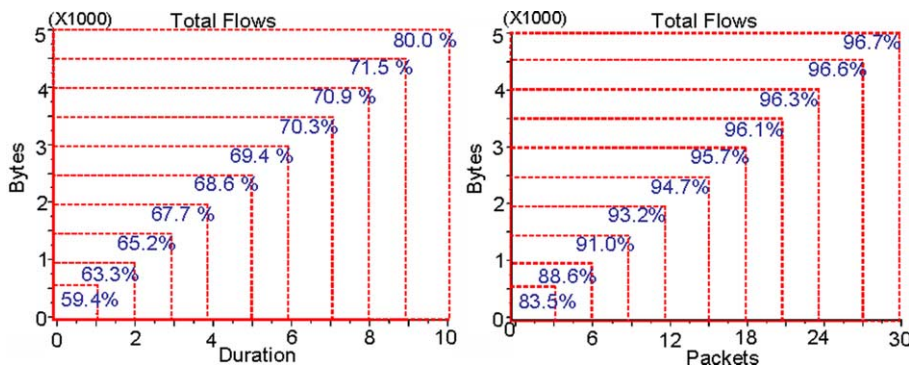


Fig. 8. Flows density in small granularity of traffic metrics.

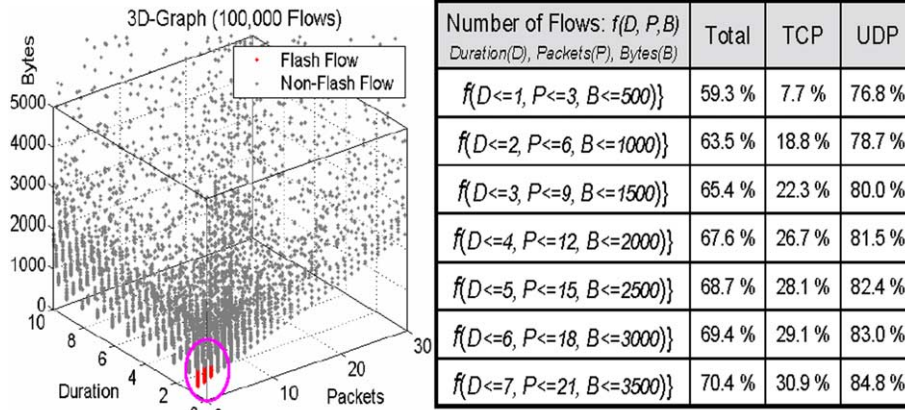


Fig. 9. Distribution of flash flows.

flash flows we specifically defined the flash flow as a flow that satisfies the following conditions:

- The duration of flash flow is less than or equal to 1 s long.
- The number of packets in flash flow is no greater than 3.
- The byte size of flash flow is no greater than 500.

The portion of flash flows is highlighted in a circle in the 3D graph of Fig. 9. In the following section, we present the detailed analysis of their characteristics and address the issues of flash flows.

5.8. Detailed analysis of flash flows

In this section, we describe comprehensive and in-depth analysis results for the traffic trace, focusing on flash flows. We investigated the impact of flash flows on the total network traffic using various analysis metrics. Finally, we tried to investigate the origin applications, which are generating these flash flows on our campus network.

5.9. Fluctuation of flows over time

Fig. 10 demonstrates the variation of flow counts over time and the effect of flash flows to the shape of graph.

The graph in Fig. 10 shows the variations of total flow counts and non-flash flow counts. The difference between the maximum and minimum values of total flow counts is 233,686 while the difference for non-flash flow counts is just 112,559. This infers that removing flash flows reduces the fluctuation of flow counts over time. When we take a close look at each minute’s flow counts, flash flows occupy more than 50% in most minutes. If we remove these flash flows, we can reduce significantly the processing overhead of the traffic monitoring system. After removing flash flows, the reduced ratios in total bytes and total packet counts are 98.56 and 92.75%, respectively. These figures are relatively high and the reduced amount is negligible. In usage based monitoring and analysis systems, the absence of flash flows improves

performance while preserving the accuracy and stability of system.

5.10. Single-flow vs. pair-flow

We defined several types of flows to investigate any hints for the influence of flash flows on the entire network link: single-flow, pair-flow, and reverse-flow. We define a single-flow as a flow having no corresponding reverse-flow and a pair-flow as a flow that has its corresponding reverse-flow. The reverse-flow of a flow is a specific flow, which satisfies the following:

An arbitrary flow is represented as fa , and its reverse flow is rfa

- The source (destination) IP of rfa is equal to the destination (source) IP of fa .
- The source (destination) port of rfa is equal to destination (source) port of fa .
- The same protocol must be used.

Table 3 shows the comparison results of single-flow and pair-flow from selected traffic trace. In the case of TCP flows, no significant difference in flow count was seen between single-flows and pair-flows. On the contrary, the number of

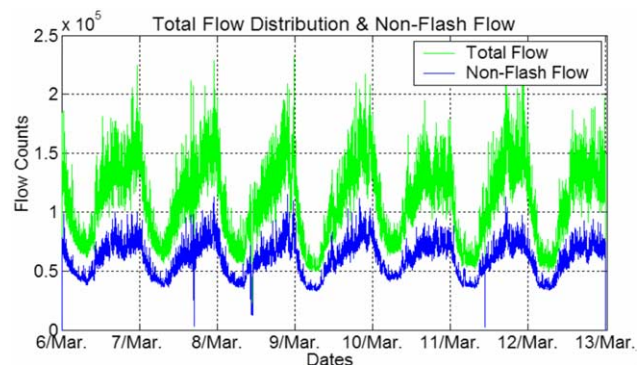


Fig. 10. Distribution of total and non-flash flow over time.

Table 3
Single-flows vs. pair-flows during 6–12 March 2004

		TCP		UDP		TCP		UDP	
Flows	Pair	104,273,000 (49.6%)		363,250,096 (62.4%)		Avg. bytes/packet	Pair	750	105
	Single	106,028,425 (50.4%)		219,561,201 (37.6%)			Single	403	371
						Avg. packet/flow	Pair	157	2
Packets	Pair	16,382,263,661 (79.1%)		915,879,767 (49.9%)		Avg. bytes/flow	Single	40	4
	Single	4,317,046,616 (20.9%)		919,452,063 (50.1%)			Pair	117910	265
						Avg. flow duration	Single	16444	1553
Bytes	Pair	12,294,918,320,736 (87.6%)		96,401,033,761 (22.0%)			Pair	78 s	7 s
	Single	1,743,559,924,046 (12.4%)		341,144,111,827 (78.0%)			Single	33 s	12 s

single-UDP flows was 1.7 times greater than the number of pair-UDP flows.

The number of packets and bytes belonging to pair TCP flows were four times and seven times greater than those of UDP flows, respectively. Yet, single- and pair-UDP flows stayed relatively still (one-to-one ratio) in the measurements using these two metrics. In other words, TCP traffic is mostly occupied by pair flows while UDP traffic is filled with single flows. We believe that this is due to nature of these two protocols. TCP guarantees a reliable connection so it can support interactive communication between users, such as downloading. On the other side, UDP is well suited for one-way data transfer (streaming) and seems to be the choice of generating abnormal traffic (e.g. network security attacks). This is a possible answer to why there are a large number of small and single-UDP flows in the campus network.

Table 4 shows the proportion of flash flows among single- and pair-flows according to their transport layer protocols. The portion of flash flows among total single-flows is 56.2%; 6% of single TCP flows and 80% of single-UDP flows are flash flows. The proportion of flash flows among TCP flows is much smaller than that of UDP flows. The portion of flash flows among total pair flows is 60.2%, which is almost the same ratio to that of single-flows. It is very interesting to observe that flash flows appear with the same ratio in both single and pair flows. This indicates that a lot of real application flows are flash flows, the details of which will be investigated in the last part of this section.

5.11. New flow occurrence

We monitor the campus network traffic every minute and count the number of new flows. We define a new flow as

Table 4
Ratio of flash flows in single and pair flows

		Total flows	Single flows	Pair flows
TCP	Total flow	210,301,425 (93%)	106,028,425 (94%)	104,273,000 (92%)
	Flash flow	14,983,561 (7%)	6,437,291 (06%)	8,546,270 (8%)
UDP	Total flow	582,811,297 (23%)	219,561,201 (20%)	363,250,096 (25%)
	Flash flow	449,650,617 (77%)	176,745,428 (80%)	272,905,189 (75%)

the first occurrence of flow in each minute; thus, every new flow is unique. Again, all the measuring processes occur in the minute range so we are dealing with new flows from a single minute.

One of the interesting facts is that the shape of the total flow graph in Fig. 11 is mostly influenced by how UDP flow count changes over time. The average number of new UDP flows per minute is greater than that of new TCP flows per minute. Unlike TCP flows, UDP flows' life cycle does not last very long. About 80% of the new UDP flows disappear in the next coming minute and a similar portion of brand new UDP flows appears again. In addition, the variation of new UDP flow occurrences over time is not steady. Therefore, the fluctuation of UDP flows graph influences the total flow distribution graph.

Fig. 12 illustrates the size of total flash flows in new flow occurrences. About 60.48% of new flows are turned out to be flash flows. The percentages of TCP and UDP flash flows over the total number of new flows are 2.77 and 57.70%, respectively. In addition, about 10.46% of total new TCP flows are flash flows while more than 78% of total new UDP flows are flash flows. It is very similar to Fig. 9 that short UDP flows influence the shape of graph and are responsible for new and short duration flow occurrences of each minute.

5.12. Port number distribution

We counted the number of flows that use port numbers below 1024 (well-known ports registered at IANA), and found that they were accountable for about 21% of the total flow counts. Over 56% of TCP flows used port numbers less

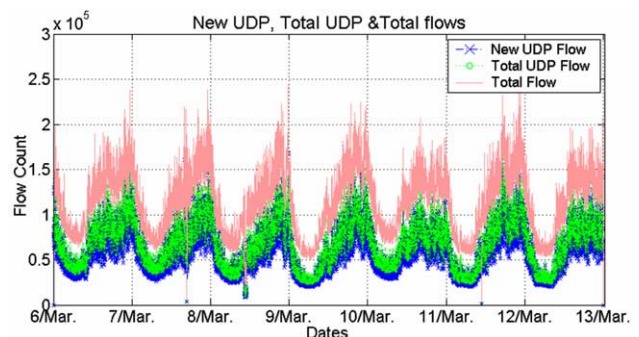


Fig. 11. New UDP flow occurrence over time.

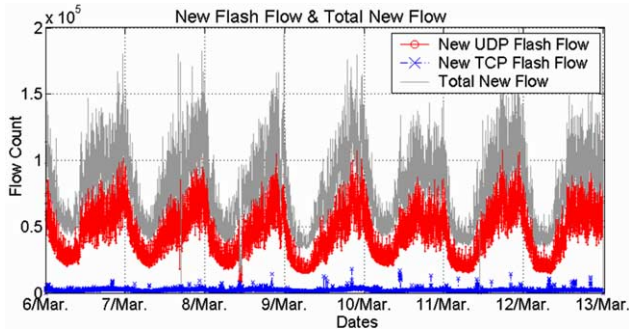


Fig. 12. New flash flow occurrence over time.

than 1024, while most UDP flows use port numbers greater than 1024. This means that a large amount of TCP flows are still generated by traditional Internet applications. However, considering UDP flows, they appeared to be newly emerging applications.

In addition, we observe that Fig. 13 has a few peaks in the distribution of total flows according to the port numbers. In order to discover the causes of these peaks we analyzed traffic

in the application layer. It turns out that several P2P file-sharing applications were responsible for those particular port usages; Overnet (4661, 4662) and Soribada (22321, 7674). The flows that fall into these applications contain a relatively small size of packets whose purpose is either a simple query or a checking message whether the peers are alive. They share similar characteristics with network security attack traffic (e.g. Internet worms); this is a concern since it becomes harder to distinguish attack traffic from normal traffic.

In terms of byte and packet counts, TCP flows overwhelm UDP flows in the range of port number beyond 1024. UDP flows spread widely over the entire range of port numbers.

To investigate the origin applications generating the flash flows, we aggregated the selected flash flows according to the port numbers, which is illustrated in Table 5. We explored the origin applications of these flows in accordance with the IANA port number list and simple testing on several end hosts. One interesting finding is that a large proportion of flash flows are concentrated on less than 50 different port numbers although being spread over the entire range of available port numbers (1 ~ 65535).

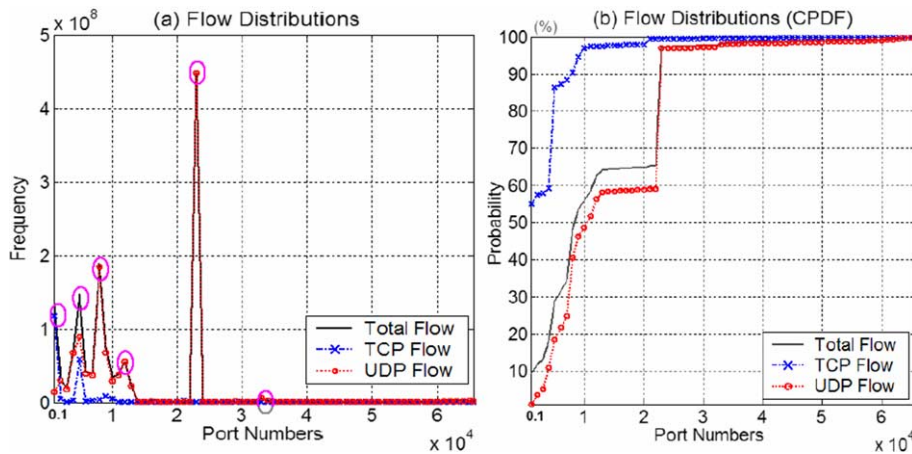


Fig. 13. Port distribution of flows.

Table 5
Distribution of flash flows according to ports

Rank	TCP			UDP		
	Port number	Cumulative flow distribution (%)	Application	Port number	Cumulative flow distribution (%)	Application
1	80	27.94%	HTTP	22321	39.55%	Soribada
2	4662	53.34%	eDonkey	7674	52.78%	Soribada
3	7000	62.34%	AFS	11518	55.01%	–
4	1863	65.88%	MSN	4672	57.18%	–
5	8404	68.29%	V-Share	53	57.82%	DNS
6	10015	70.28%	–	5383	58.15%	–
7	20168	72.08%	–	10106	58.44%	–
8	3128	73.74%	Active API	10925	58.74%	–
9	1080	75.40%	–	9915	59.02%	–
10	6129	76.92%	–	11282	59.30%	–
20	4661	87.30%	eDonkey	10029	61.19%	–
30	3531	90.60%	–	5991	63.85%	–
Total		100.00%			100.00%	

In the case of TCP flash flows, 76% of them use only ten different port numbers. The HTTP traffic generates the largest number of flash flows; however, it might have contained abnormal traffic by invalid use of port 80. A majority of them do not have their corresponding reverse flows. File sharing and instant messaging applications, such as eDonkey, V-Share and MSN, also generate a significant amount of flash flows in the network next to HTTP. What separates these newly emerging applications from HTTP is that they communicate with multiple peers to complete their services and rely on light-weighted and instant packets for their communication needs. Considering UDP flash flows, the flash flows over UDP are also spread over the entire range of port numbers with much higher hit counts than TCP. Because these ports are not registered at the IANA port list, it is not easy to discover their origin applications. The port number based application identification method has some limitation to determine these new UDP flows.

In this port number distribution, we observe some symptoms of network or host scanning from DoS/DDoS or the Internet worms. For example, we spotted a large amount of flash flows departing from or reaching to certain end hosts with a single port number and fixed packet size. Along with some P2P applications that we discovered, the abnormal traffic was one of the major causes generating large portions of flash flows.

6. Summary

The following are some findings about recent IP traffic generated in the campus network from the various flow-based analyses.

- The byte count and packet count of TCP traffic is much larger than those of UDP traffic. However, the flow count of TCP traffic is two times smaller than UDP traffic.
- Over 50% of flows are occupied by those with three packets, less than 500 b size, and less than 1 s. Most of them are UDP flows. We call these short and small size flows, flash flows.
- Most flash flows are UDP flows. In our investigation, only 7.7% of TCP flows are flash flows, while 76.8% of UDP flows are flash flows.
- Flash flows mainly influence the time-of-day fluctuation in the number of concurrent TCP and UDP flows in time-series graph.
- Most flash flows are generated by peer-to-peer applications and some abnormal traffic such as DoS/DDoS attacks and Internet worms.

The flash flows may be less important in traditional traffic monitoring systems. The sampling mechanism introduced from the IETF PSAMP working group [13], sFlow [29], and the research result by Estan et al. [14] remove the flash flows. However, these flows are vital in the newly emerging traffic

analysis areas, such as P2P traffic analysis and abnormal traffic detection. As long as there is no strong reason to investigate all the flash flows in terms of accuracy, it is recommended that a flexible flow exporting scheme should be developed to improve the performance of traffic analysis systems. Therefore, we should decide how to handle these flash flows according to the purpose of traffic analysis and find a suitable method to reduce the number of flash flows without giving up the original purpose.

7. Concluding remarks

A number of current systems for traffic monitoring and analysis focus on flow-based investigation. One of the key assets of these systems is to compress a significant amount of packet data into flows. However, the compression ratio is highly variable in the recent network environment due to the increasing use of P2P file sharing applications and the frequent appearances of abnormal traffic. In this paper, we studied the characteristics of the IP traffic from the perspective of flows and found that the flash flows with short-time durations and small size are significant in the consideration of traffic analysis system performance.

It is not clear under what circumstances the flash flows should be taken into consideration or discarded by the network monitoring systems where they have different purposes. However, we believe that the different consideration on flash flows in the development of traffic analysis system should be taken according to the analysis purpose and this consideration will yield significant improvement in the system performance.

In future work, we plan to study the features of flash flows in more detail, such as application-level characteristics of flash flows and the behavior of flash flows of abnormal traffic. By conducting numerous experiments, we intend to develop an efficient method to handle flash flows for various purposes of traffic analysis. If we combine the flow property with other mechanisms (like clustered architecture and sampling method) we can develop an improved traffic analysis system.

References

- [1] Se-Hee Han, Myung-Sup Kim, Hong-Taek Ju, J.W. Hong, The Architecture of NG-MON: A Passive Network Monitoring System, DSOM 2002, Montreal Canada, Oct 2002 p. 16–27.
- [2] Luca Deri, Ntop, <http://www.ntop.org>.
- [3] Daniel W. McRobb, cflowd design, CAIDA, Sep. 1998.
- [4] J.Quittek, T. Zseby, B. Claise, K.C. Norsth, IPFIX Requirements, Internet Draft, <http://norseth.org/ietf/ipfix/draft-ietf-ipfix-architecture-00.txt>.
- [5] CAIDA, Preliminary Measurement Spec for Internet Routers, <http://www.caida.org/tools/measurement/measurementspec/>.
- [6] Cisco, NetFlow Services and Applications, White Paper, http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neftct/tech/napps_wp.htm.
- [7] IETF ipfix (IP Flow Information Export) WG, <http://www.ietf.org/html.charters/ipfix-charter.html>.

- [8] S. Saroiu, K.P. Gummadi, R.J. Dunn, S.D. Gribble, H.M. Levy, An Analysis of Internet Content Delivery Systems, OSDI 2002, Boston, MA, Dec. 2002.
- [9] Kun-chan Lan, A. Hussain, D. Dutta, Effect of Malicious Traffic on the Network, PAM 2003, San Diego, California, Apr. 2003.
- [10] Argus, <http://www.qosient.com/argus/index.htm>.
- [11] N. Brownlee, Traffic Flow Measurement: Experiences with NeTraMet, IETF RFC2123, Mar. 1997.
- [12] Ahang Shiyong, Wu Chengrong, Guw Wei, Network monitoring in broadband network Proc. of Web Information Systems Engineering, Vol. 2, Dec. 3–6 2001.
- [13] IETF psamp (Packet Sampling) WG, <http://www.ietf.org/html.charters/psamp-charter.html>.
- [14] C. Estan, G. Varghese, New directions in traffic measurement and accounting Proc. of the IMW2001, San Francisco, CA, November 2001.
- [15] Internet2, <http://netflow.internet2.edu/weekly/>, 2003.
- [16] A. Gerber, J. Houle, H. Nguyen, M. Roughan, S. Sen, P2P the gorilla in the cable NCTA 2003 National Show, Chicago, IL, June 8–11 2003.
- [17] Graffiti port number list, <http://www.graffiti.com/services>.
- [18] J. van der Merwe, R. Caceres, Yang-hua Chu, C. Sreenan, mmdump—tool for monitoring internet multimedia traffic ACM Computer Communication Review, October 2000 p. 48–59.
- [19] Hun-Jeong Kang, Myung-Sup Kim, James.Won-Ki Hong, A method on multimedia service traffic monitoring and analysis DSOM 2003, Heidelberg, Germany, October 2003 p. 93–105.
- [20] T.S. Choi, C.H. Kim, S.H. Yoon, J.S. Park, H.S. Chung, B.J. Lee, H.H. Kim, T.S. Jeong, Rate-based internet accounting system using application-aware traffic measurement APNOMS 2003, Fukuoka, Japan, October 1–3 2003 p. 404–415.
- [21] R. Poortinga, R. van de Meent, A. Pras, Analysing campus traffic using the meter-MIB PAM2002, Mar. 25–27 2002.
- [22] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, C. Diot, Packet-level traffic measurements from the sprint IP backbone IEEE Network 2003.
- [23] Sharad Agarwal, Chen-Nee Chuah, Supratik Bhattacharyya, Christophe Diot, “The Impact of BGP Dynamics on Intra-Domain Traffic,” Sprint ATL Research Report No. RR03-ATL-111377, Sprint ATL, Nov. 2003.
- [24] S. Sen, J. Wang, Analyzing peer-to-peer traffic across large networks Proceedings of IMW2002 2002.
- [25] R. Bhagwan, S. Savage, G. Voelker, Understanding availability Proceedings of IPTPS 2003, Berkeley, CA, February 2003.
- [26] P.K. Gummadi, S. Saroiu, S. Gribble, A measurement study of Napster and Gnutella as examples of peer-to-peer file sharing systems, Multimedia Systems 9 (2) (2002) 2002.
- [27] N. Leibowitz, A. Bergman, R. Ben-Shaul, A. Shavit, Are File Swapping Networks Cacheable? Seventh International Workshop on Web Content Caching and Distribution (WCW), Boulder, Colorado, Aug. 14–16 2002.
- [28] G. Iannaccone, C. Diot, I. Graham, N. McKeown, Monitoring Very High Speed Links Proc. of IMW, San Francisco, November 2001.
- [29] P. Phaal, S. Panchen, N. McKee, InMon Corporation’s sFlow: A Method for Monitoring Traffic in Switched and Routed Networks IETF RFC 3176, September 2001.