Analysis of a Distributed Denial-of-Service Attack

Ka Hung HUI and OnChing YUE
Mobile Technologies Centre (MobiTeC)
The Chinese University of Hong Kong

Abstract

DDoS is a growing problem in cyber security. One DDoS defense technique actively studied by researchers is on-line packet attribute analysis followed by selective packet filtering. In order to evaluate the effectiveness of this technique, we have analyzed the packet traffic data collected at the routers in two sites: a university department network (16,800,000 packets/hr) and an ISP backbone network (23,500,000 packets/hr) during a DDoS attack. In this report, we first summarize the system model which is the basis for the approach of packet filtering. Then we describe our technique for analyzing the data collected by the NetFlow measurement system. Finally, we present the results on the histograms of the different packet attributes under normal and attack scenarios. We observe that there are significant differences in the histograms under different scenarios, so that attack detection based on packet attribute analysis will be effective. Moreover, we note that there is a ramp up period (several minutes) of attack traffic volume, which should allow enough time for the selective packet filtering procedure to be implemented before serious damage is done to the resource under attack.

1. Introduction

Distributed Denial-of-Service (DDoS) is one type of cyber attacks in which the victim receives a large amount of attack packets coming from a large number of hosts. As a result, the victim will be overloaded and eventually it will be unable to perform any normal functions.

Currently, any counter measures are done manually. When an attack is reported, offline traffic analysis will be carried out to identify the possible attacks. After identification, new access controls will be set up to filter the attack packets.

An example of such procedure is currently used by iAdvantage, a local ISP. MRTG (Multi Router Traffic Grapher) is used to monitor the traffic load by generating HTML pages containing graphical images which provide a live visual representation of the traffic. [6] If any anomaly is observed, data in NetFlow database [1] will be used to check the packet attributes, like IP, flow count, packet rate, etc. The same set of data collected 15 minutes before will be served as the baseline for comparison. If any customers are identified as the source or

destination of attacks, the switch port associated with the customers will be closed down manually. And the ISP will contact the customers to find out the causes of the attacks and the methods to tackle the attacks.

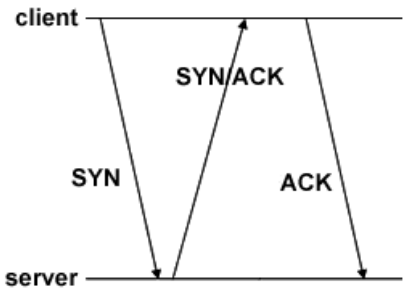
The major disadvantage of this approach is that the response time may be too long. Damages may occur before new access controls are established, or even before the detection of attacks.

To tackle the issue of response time, we propose a new method to deal with DDoS: automatic detection of attack traffic. If the network can detect attacks automatically, the response time may be shortened and damages may be reduced.

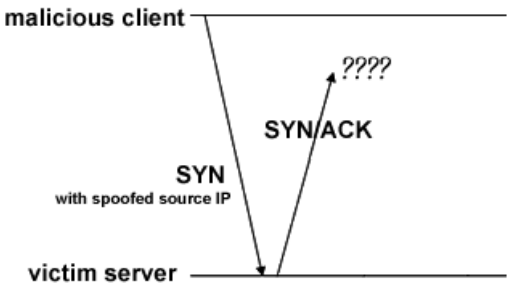
To establish the feasibility of our approach, patterns of normal traffic data and attacking traffic data are obtained. Then the distributions of packet attributes in normal condition and attacking condition are obtained and compared to find out the deviation of attributes under attack from normal condition. If any anomaly is found, it may facilitate the identification of attack packet signature.

1.1 Anatomy of a DDoS TCP SYN Flood Attack

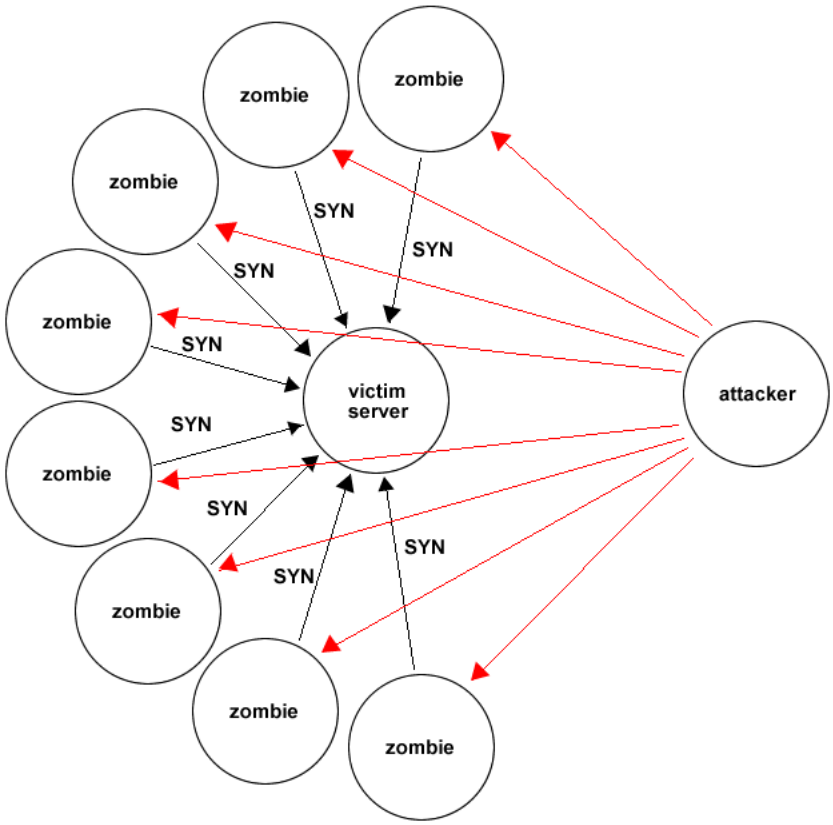
In this section we describe the type of DDoS attack captured in the measured data from the ISP backbone network. The establishment of a TCP connection typically requires the exchange of three IP packets between two machines in an interchange known as the **TCP Three-Way Handshake.** [8]



In a traditional SYN Flood attack, a malicious client sent a SYN packet with a fraudulent source IP address. As a result, the SYN/ACK packet sent by the victim server will not get a reply as shown below.



In a DDoS TCP SYN Flood attack, the malicious client first infects a group of innocent clients called “zombies” and then launches a coordinated attack on the victim.



1.2 System Model of DDoS Detection

In this section, we define system model in terms of the packet attributes of interest to DDoS and their distributions, and describe how the variations in the packet histograms can be used to detect the onslaught of a DDoS attack and filter out the undesirable packets.

We model the packet stream arriving at a router as a stochastic process $\{\vec{X}(n)\}$, where $\vec{X} = [X_1, X_2, \cdots, X_K]$ is a vector of K random variables associated with the n^{th} packet. The

random variables are the attributes of an IP datagram such as packet length, protocol, source and destination addresses and port numbers. For example, if X_1 is the protocol field in the IP header, then the possible values are 1 (ICMP), 2 (IGMP), 6 (TCP), 17 (UDP), etc. Assuming that the system is stationary, we shall define the joint distribution of the attributes as $P(X_1, X_2, \cdots, X_K)$. The basic idea behind our DDoS detection is that the attribute distribution under normal and attack scenarios is different.

Which set of attributes is sufficient for DDoS defense will depend on the nature of the attack. (We will comment on this more after showing the experimental results.) For example, if we know that a particular resource with destination address 196.xxx.yyy.zzz is being attacked, we can monitor the destination address of all packets and filter or throttle those packets with this attribute value. Therefore, the challenge is to identify the attribute to monitor and decide on the suspicious attribute value(s).

To illustrate our theory, we shall focus on one attribute, denoted as X, of the packets arriving at the router. Let $f(x)$ be the probability density function of X for the normal packets and $g(x)$ be the density of the same attribute for attack packets. Under normal conditions, the arrival rate of packets is λ_a, and the density $p(x)$ for X of the arriving packets is $p(x) = f(x)$. When the network is under attack, the aggregate arrival rate is $\lambda = \lambda_a + \lambda_b = (\alpha + \beta)\lambda$, and $p(x) = \alpha f(x) + \beta g(x)$. In the following we shall consider different algorithms of discarding packets. Let $P_1 = P[discard \mid normal]$ denote the probability of discarding normal packets and $P_2 = P[retain \mid attack]$ the probability of retaining attack packets.

2. NetFlow Database Analysis

A network flow is defined as a unidirectional sequence of packets between given source and destination endpoints. [1] The NetFlow database saves network traffic by inspecting and storing flow records. The database consists of one header, and varying number of flow records. [4]

The information contained inside the header includes:

- Version number: the version that the NetFlow database is using. Currently, versions 5 and 7 are used in our analysis;
- Total number of flow records;
- Router boot time;
- Current time since 0000 UTC 1970 in milliseconds;
- Number of residual nanoseconds since 0000 UTC 1970;
- Sequence counter;

For NetFlow version 7, additional header information is included:

- Type of flow-switching engine;
- Slot number of the flow-switching engine.

Each flow record is uniquely identified by the following seven attributes:

- Source IP;
- Destination IP;
- Source port;
- Destination port;
- Layer 3 protocol byte;
- TOS byte;
- SNMP input interface index.

Besides the seven attributes, each flow record also contains the following information for NetFlow versions 5 and 7:

- Number of packets & bytes in a flow;
- Flow start time & end time;
- SNMP output interface index;
- TCP flags;
- Routing Information (Next hop router IP, Source & Destination subnet mask, Source & Destination AS number).

For NetFlow version 7, additional information is provided:

- Shortcut mode flags;
- Shortcut router IP.

2.1 Techniques in Analyzing of NetFlow Data

The data in NetFlow database is preprocessed by flow-tools. [3] To save storage, the data is compressed using zlib [7] before exporting to a data file. In order to decode the data correctly, zlib version 1.0.4 or greater should be used to decompress the data file first.

The histograms are built using linked-list implementation. Comparing with using array, this has the advantage that if a particular value does not exist, it is not necessary to store this value, thus saves memory space. However, this approach is only suitable if there is only a small portion of

values contain non-zero frequency. Otherwise, the linked list obtained would be too long, the time needed for traversal and thus the time for detecting anomaly would be too long.

A sorted linked-list is used in building histograms. By doing so, it is not necessary to traverse the whole linked-list to check if the attribute value exists or not. An example is shown in Fig.1.

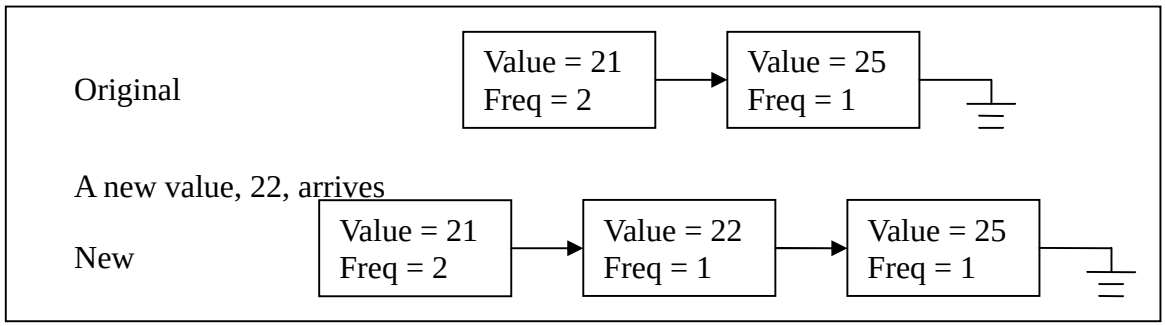


Fig 1. Updating a sorted linked-list.

When linked-list implementation is used, the most time-consuming part is dereferencing. For 8-bit attribute values, there are 256 possible values, one linked-list is sufficient without significant delay in processing (about 2 minutes is needed for decompressing the 9M-file, building the histogram and writing to a file). However, for 16-bit attribute values, there are 65536 possible values, using one linked-list would introduce significant delay in processing (the program cannot terminate after 10 minutes). Currently, 16 linked-lists are used simultaneously, with the first one store values of 0-4095, the second 4096-8191, and so on (about 3 minutes is needed for the whole process).

2.2 Measurement Results of 2 Networks

Network 1: IE Network

First, we show the invariance nature of the distribution of packet attributes in normal conditions. The invariance nature of the distribution of packet attributes in normal condition serves as the baseline for comparison with the distribution of packet attributes in attacking condition.

The two NetFlow data files used were collected in the Department of Information Engineering, CUHK on 23 May, 2004. The normalized frequency of the packet attributes are calculated in one-hour intervals. Both files are of version 7. The details of the files are shown in Table 1.

NetFlow File	ft-v07.2004-05-23.180000+0800	ft-v07.2004-05-23.190000+0800
Time of collection	23 May, 2004 18:00-19:00	23 May, 2004 19:00-20:00

Number of flows	612,735	601,698

Table 1. Details of the NetFlow data files in normal condition.

The packet attributes concerned are source port, destination port, layer 3 protocol byte, average packet length and TOS byte. The distributions of various packet attributes are shown in Figure 2-6 respectively. As shown in the figures, the same peaks appear in the histograms. Also, the variations of the normalized frequency of the packet attributes are within a few percentage of the total number of packets.

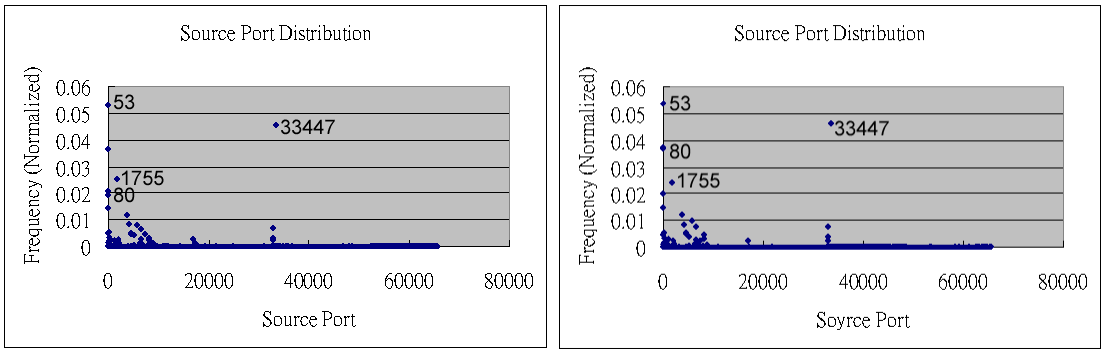


Figure 2. Distributions of source port for ft-v07.2004-05-23.180000+0800 (left) and ft-v07.2004-05-23.190000+0800 (right).

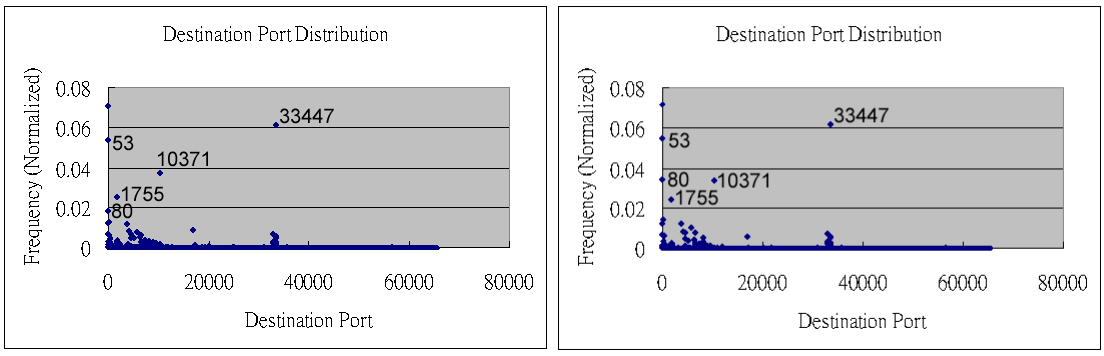


Figure 3. Distributions of destination port for ft-v07.2004-05-23.180000+0800 (left) and ft-v07.2004-05-23.190000+0800 (right).

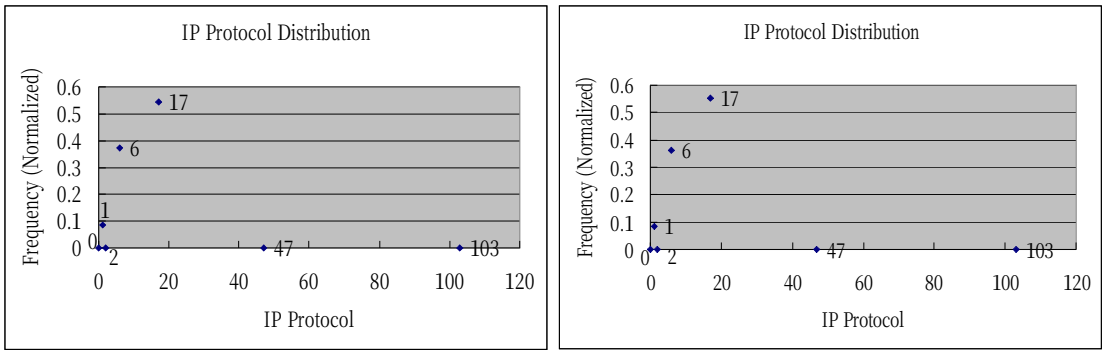


Figure 4. Distributions of layer 3 protocol byte for ft-v07.2004-05-23.180000+0800 (left) and ft-v07.2004-05-23.190000+0800 (right).

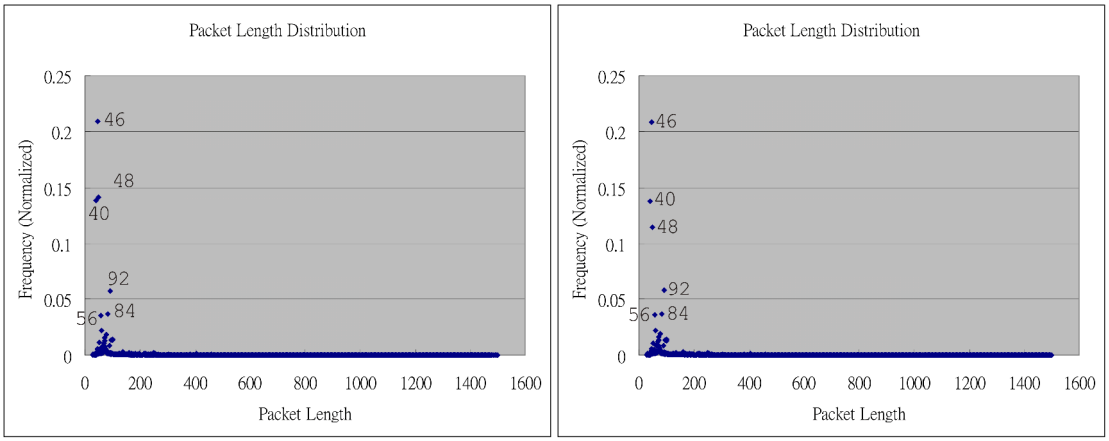


Figure 5. Distributions of average packet length for ft-v07.2004-05-23.180000+0800 (left) and ft-v07.2004-05-23.190000+0800 (right).

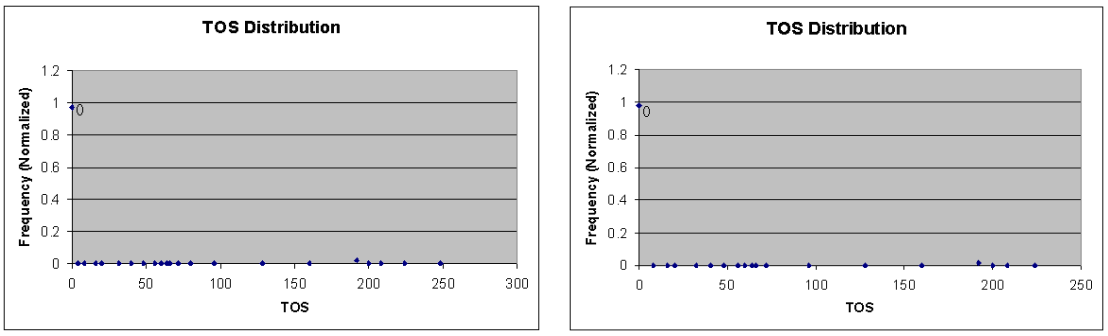


Figure 6. Distributions of TOS byte for ft-v07.2004-05-23.180000+0800 (left) and ft-v07.2004-05-23.190000+0800 (right).

Next, we show the differences between the distribution of packet attributes in normal condition and attacking condition. The NetFlow data file containing attack traffic data was collected on 9 June, 2004. The attack data was generated as a result of SQL slammer attack. Both files are of version 7. The details of the files are shown in Table 2.

NetFlow File	ft-v07.2004-05-23.180000+0800	ft-v07.2004-06-09.060001+0800
Time of collection	23 May, 2004 18:00-19:00	09 June, 2004 06:00-07:00
Number of flows	612,735	429,669

Table 2. Details of the NetFlow data files in normal condition and the condition under attack.

The distributions of source port, destination port, layer 3 protocol byte, average packet length and TOS byte in attacking condition are shown in Figure 7-11 respectively. The corresponding distributions in normal condition are included for comparison. The following observations are obtained when comparing the histograms:

- The peaks originally appearing in normal condition may be suppressed during attack, e.g., source port 53 and 80.
- The peaks originally appearing in normal condition may be enhanced, e.g., IP protocol 17 (UDP).

- New peaks may appear as a result of the attack, e.g., destination port 1434.
- There may be no significant differences between the histograms, e.g., distribution in TOS byte.

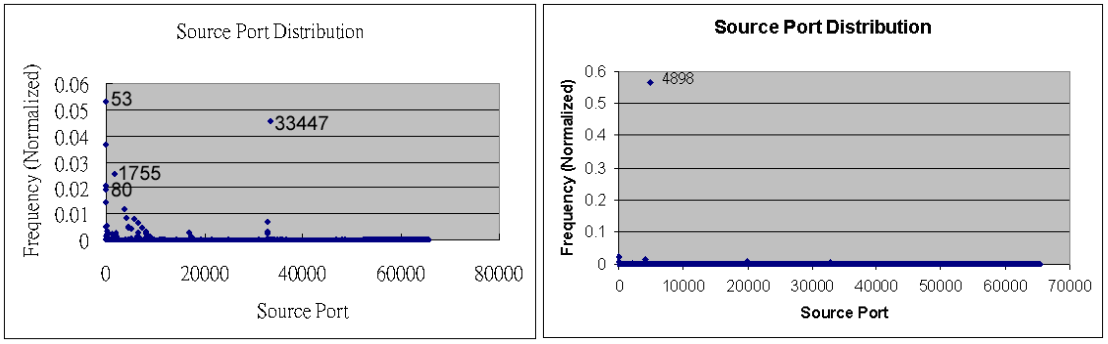


Figure 7. Distributions of source port for ft-v07.2004-05-23.180000+0800 (left) and ft-v07.2004-06-09.060001+0800 (right).

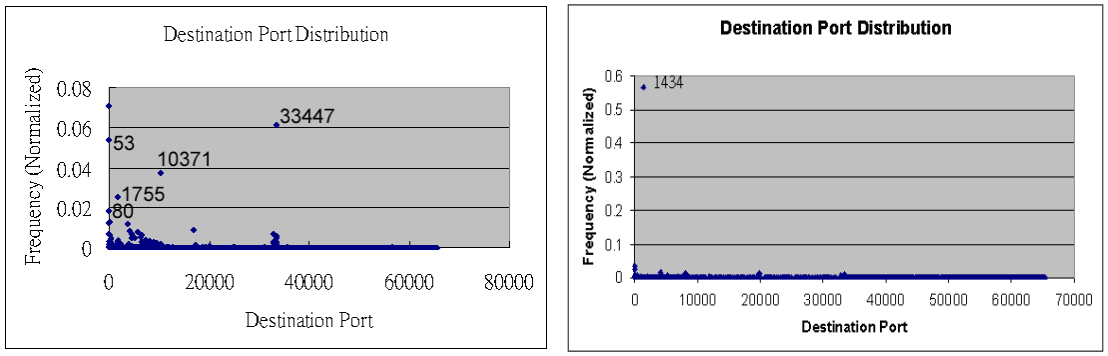


Figure 8. Distributions of destination port for ft-v07.2004-05-23.180000+0800 (left) and ft-v07.2004-06-09.060001+0800 (right).

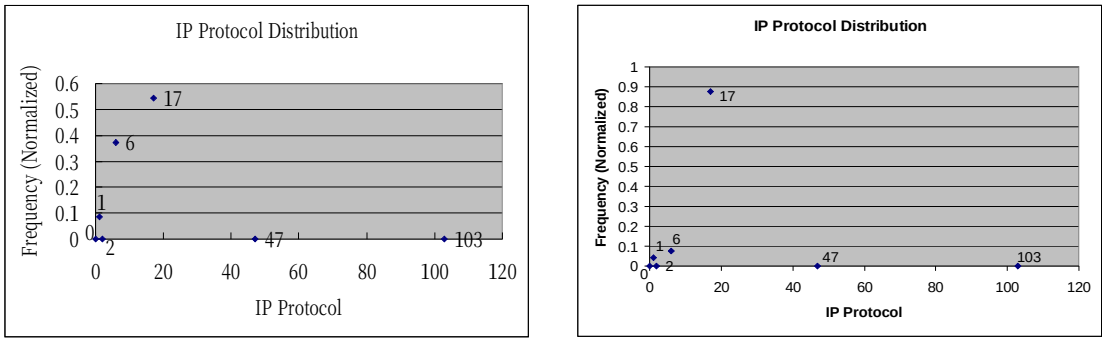
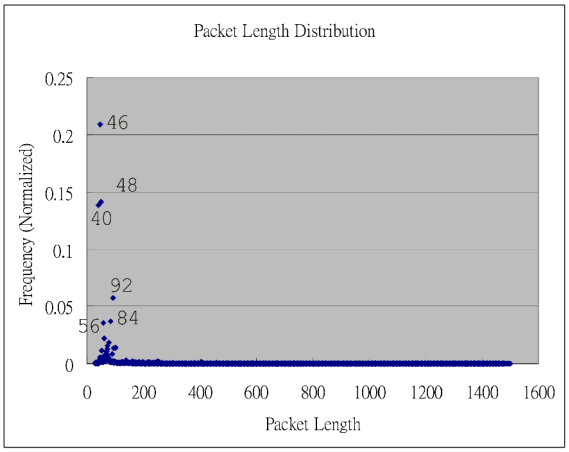


Figure 9. Distributions of layer 3 protocol byte for ft-v07.2004-05-23.180000+0800 (left) and ft-v07.2004-06-09.060001+0800 (right).



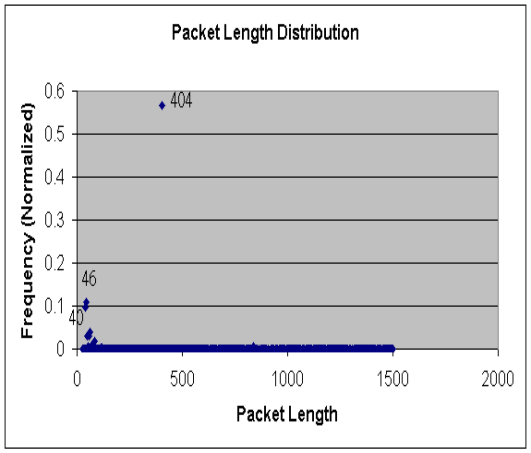
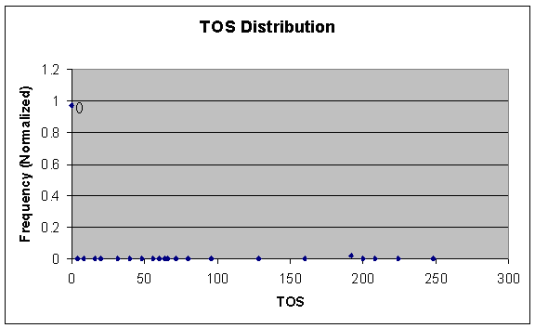


Figure 10. Distributions of average packet length for ft-v07.2004-05-23.180000+0800 (left) and ft-v07.2004-06-09.060001+0800 (right).



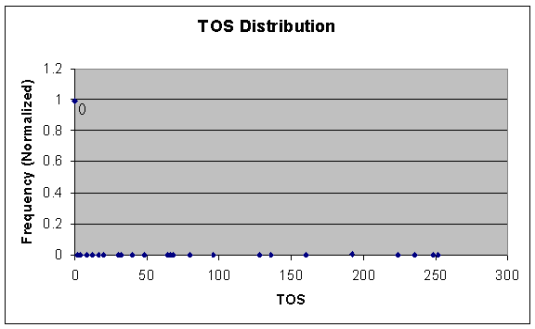


Figure 11. Distributions of TOS byte for ft-v07.2004-05-23.180000+0800 (left) and ft-v07.2004-06-09.060001+0800 (right).

The identified packet signature for the attack is shown in Table 3. The attributes identified here conforms to the attack generated by SQL slammer. [2]

Packet Attribute	Attribute Value
Source IP	137.189.98.1
Layer 3 Protocol	17
Source Port	4898
Destination Port	1434
Byte Count	404
Packet Count	1
TOS Byte	0x00

Table 3. Signature of the attack packet

Network 2: iAdvantage Network

The NetFlow data files provided by iAdvantage were collected on 5-minute basis. They were collected on 6 April, 2004. All the files are of version 5. We observe that the number of flows captured in each 5-minute interval increases gradually from 310,000 to 470,000. This shows that the amount of traffic depends on the time of the day.

The major difference between the data in the two networks is due to the distance (in terms of hop count) between the router and the victim. For the IE network, the router is close to the victim, so the attack packets occupy a significant portion in the traffic processed by the router. Attacks are easy to identify. On the other hand, the router in iAdvantage is a backbone router. The router is much farther away from the victim. The proportion of the attack packets in the total traffic in backbone router is much smaller than that in the router close to the victim. Although peaks generated by attacks may still be observed, the value of the peaks may be much smaller and it may be more difficult to observe.

The identified attack packet signatures are shown in Table 4-5. The attack was initiated by a lot of hosts sending packets to destination IP 202.xx.yyy.37 and destination port 27015, requesting TCP connection setup. The host 202.xx.yyy.37 did not have enough resources do handle all the connection requests, so it sent RST packets to close the connections. The port 27015 is used for multi-player gaming.

Packet Attribute	Attribute Value
Destination IP	202.xx.yyy.37
Layer 3 Protocol Byte	6
Destination Port	27015
Bytes per Packet	48
TOS Byte	0x00
TCP Flag	0x02 (SYN)

Table 4. Signature of the attack packet 1 (time duration: 09:19:31-12:59:37)

Packet Attribute	Attribute Value
Source IP	202.xx.yyy.37
Layer 3 Protocol Byte	6
Source Port	27015
Bytes per Packet	40
TOS Byte	0x00
TCP Flag	0x14 (ACK, RST)

Table 5. Signature of the attack packet 2 (time duration: 09:19:26-09:50:25)

3. Properties of DDoS observed in iAdvantage Network

In the analysis of the data provided by iAdvantage network, some properties of DDoS are observed. These are illustrated together with their inferences on intrusion detection algorithms as follows:

3.1 Ramp up period

We observe that for the attack packets to become easily detectable, a certain period of time is needed for the attack packets to occupy a significant portion in the traffic processed by a router. We label this period as “ramp up period”. The ramp up period lasts for about 2 minutes in this case. As shown in the figures below, the normalized frequency of source port 27015 rises from 0.1% to 4.5% of the total traffic when the attack began. The normalized frequency of destination port 27015 increased to a greater extent, from 0.04% to 14%, and then fell back to 7.5%.

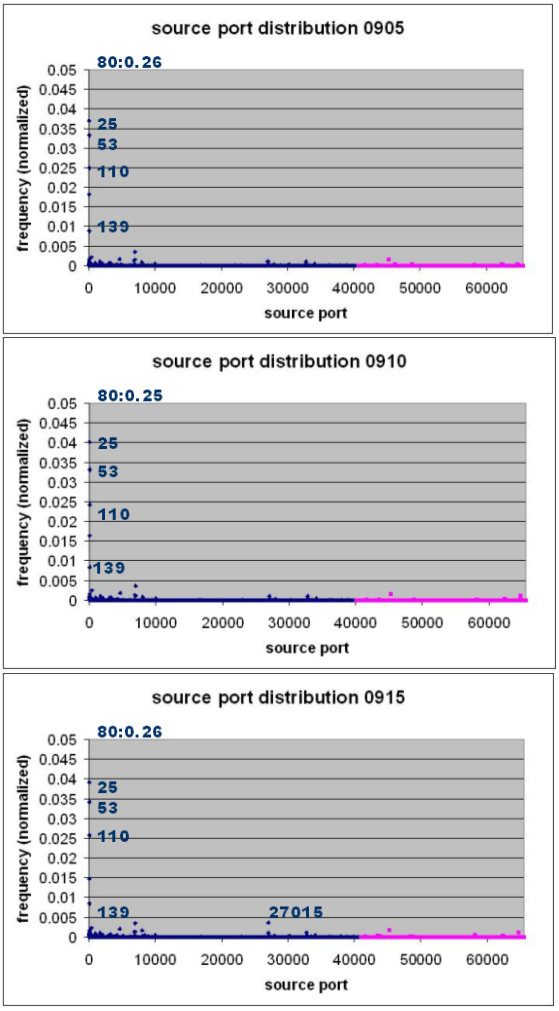


Figure 12. Variation in per-flow source port distribution in 0905-0955.

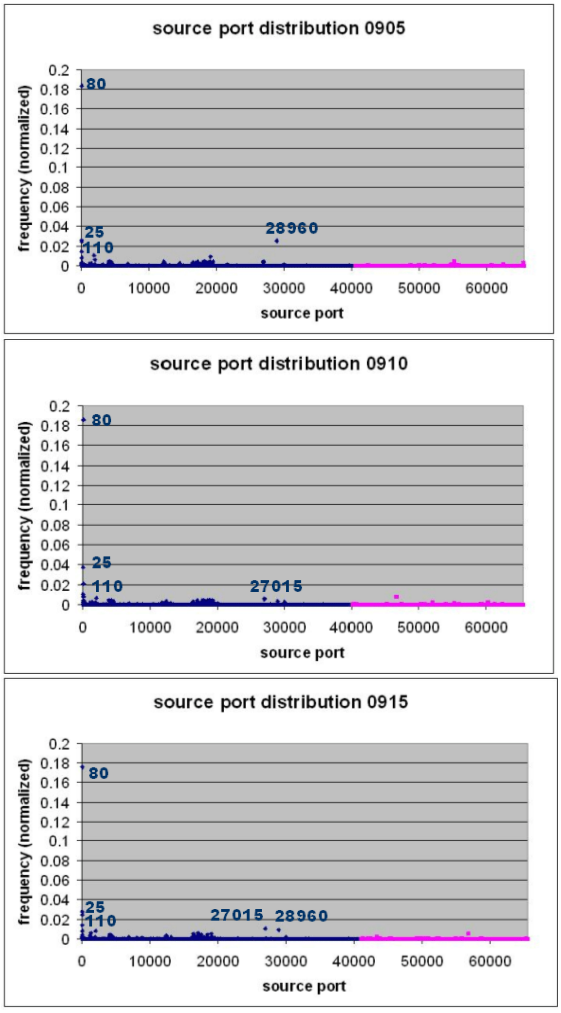


Figure 13. Variation in per-packet source port distribution in 0905-0955.

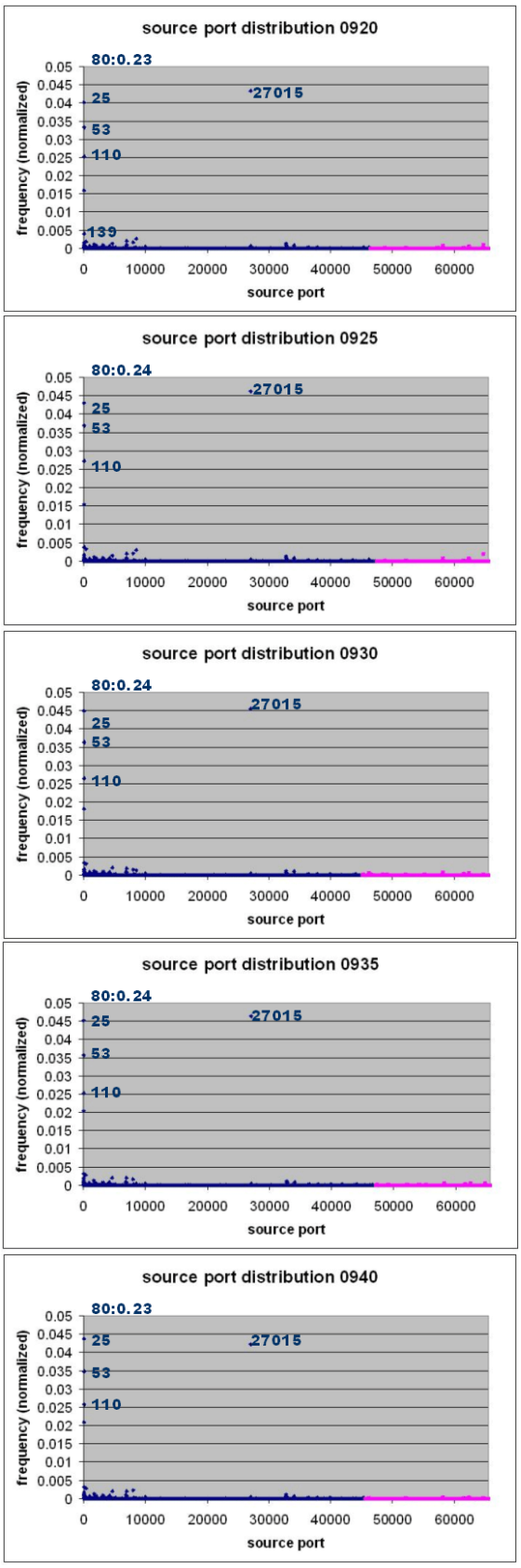


Figure 14. Variation in per-flow source port distribution in 0905-0955 (continued).

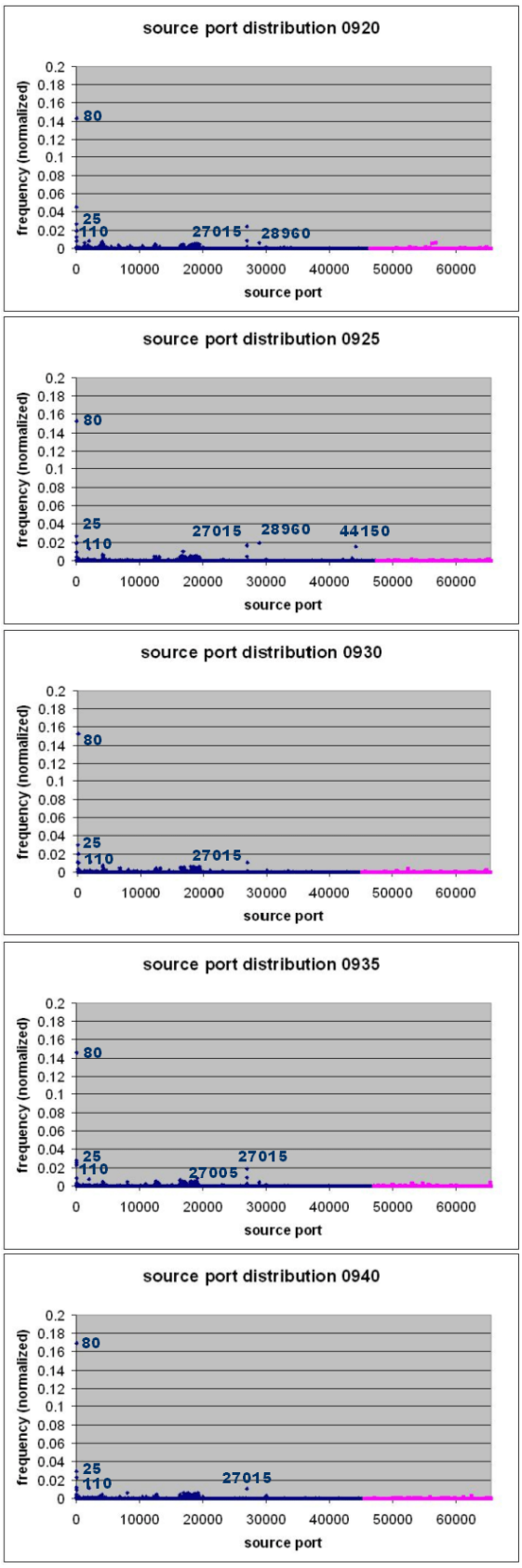
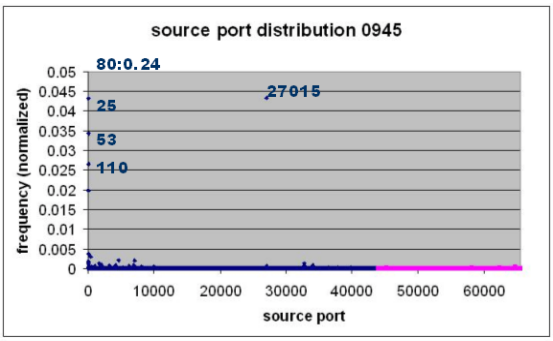
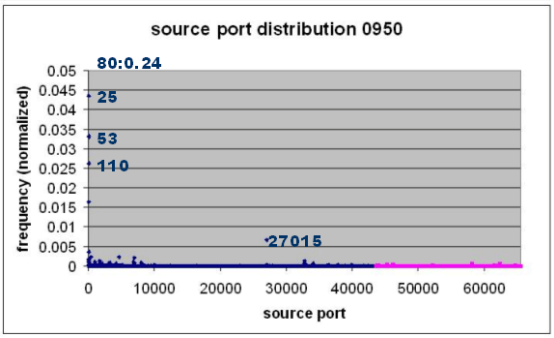


Figure 15. Variation in per-packet source port distribution in 0905-0955. (continued).





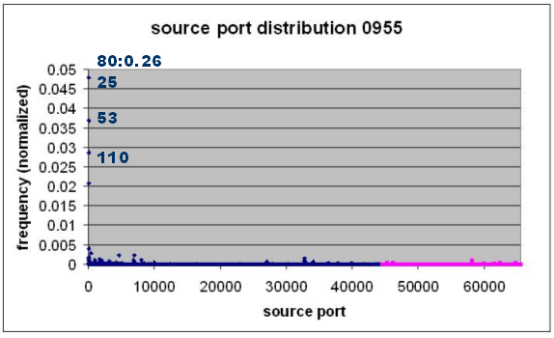
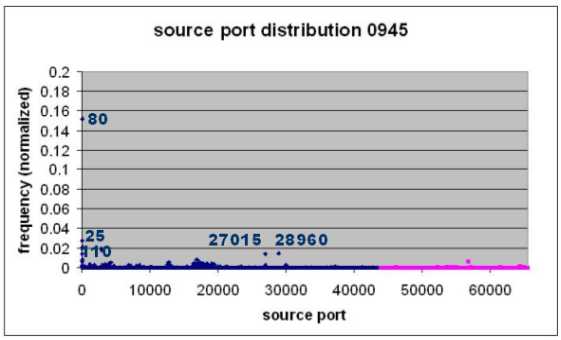
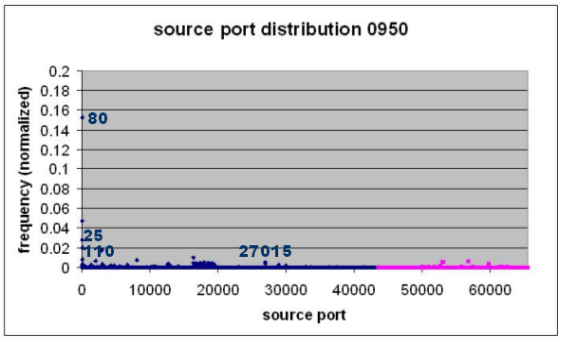


Figure 16. Variation in per-flow source port distribution in 0905-0955 (continued).





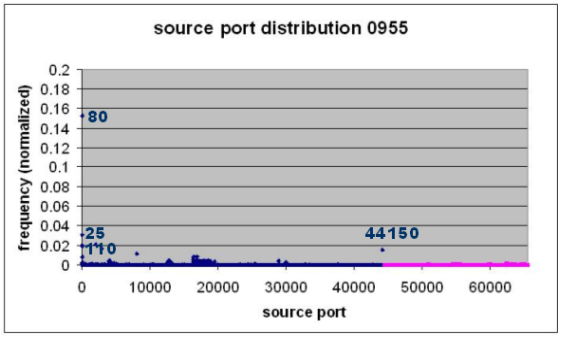


Figure 17. Variation in per-packet source port distribution in 0905-0955. (continued).

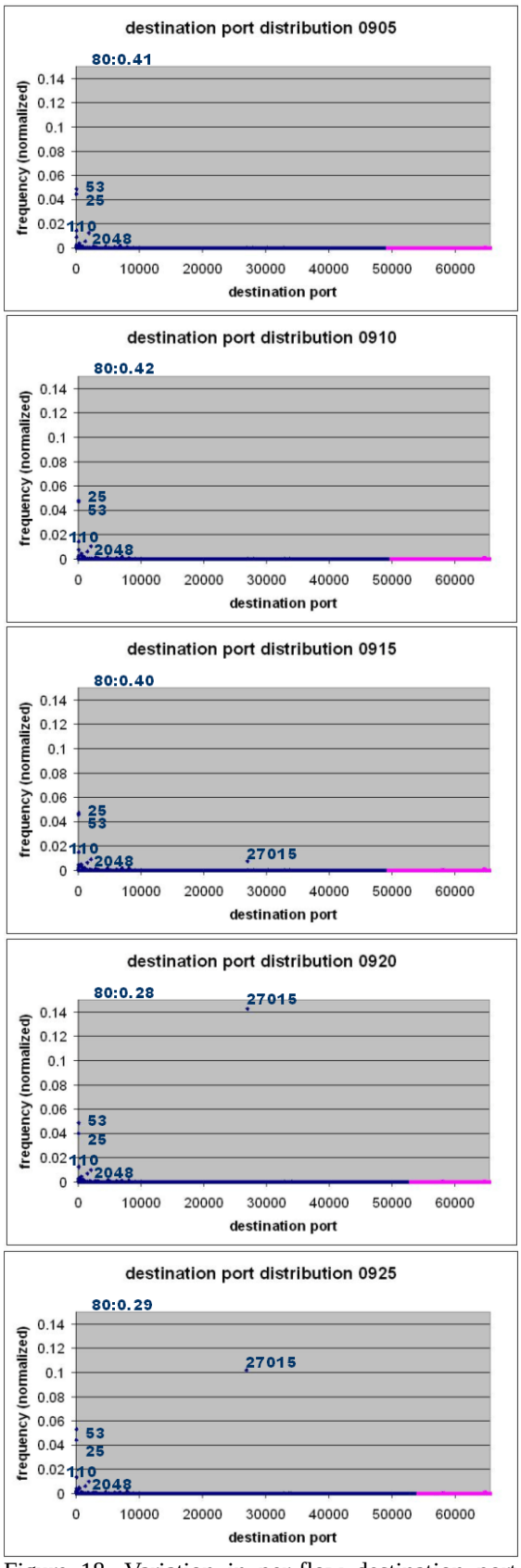


Figure 18. Variation in per-flow destination port distribution in 0905-0955.

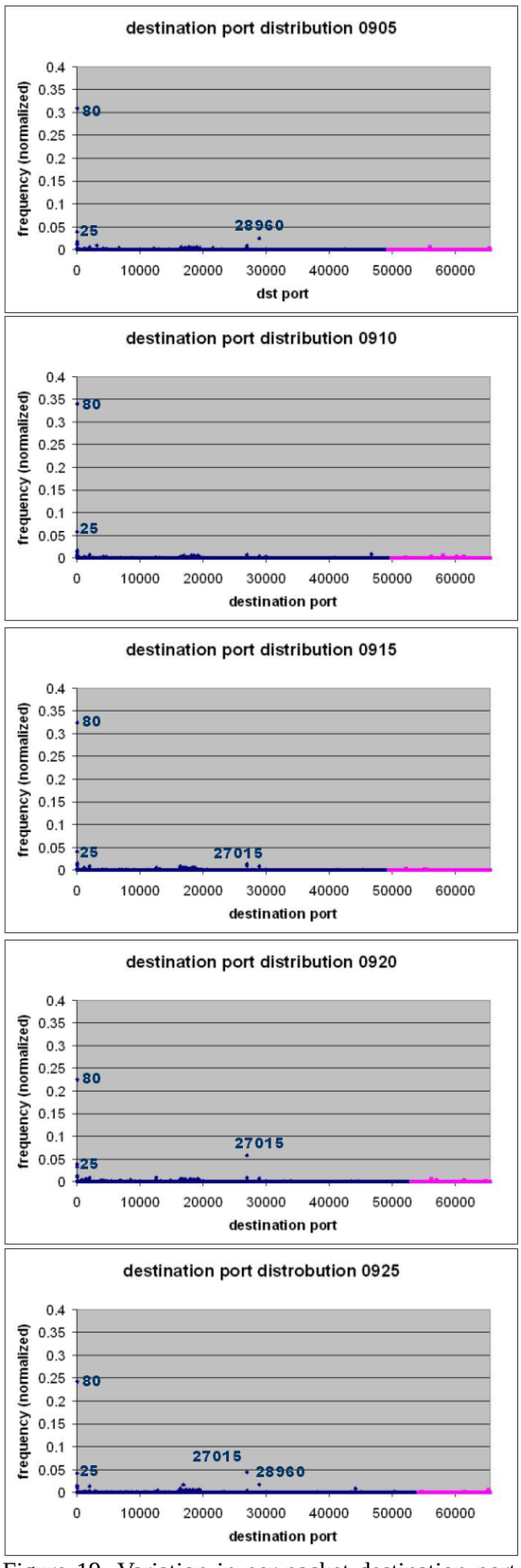


Figure 19. Variation in per-packet destination port distribution in 0905-0955.

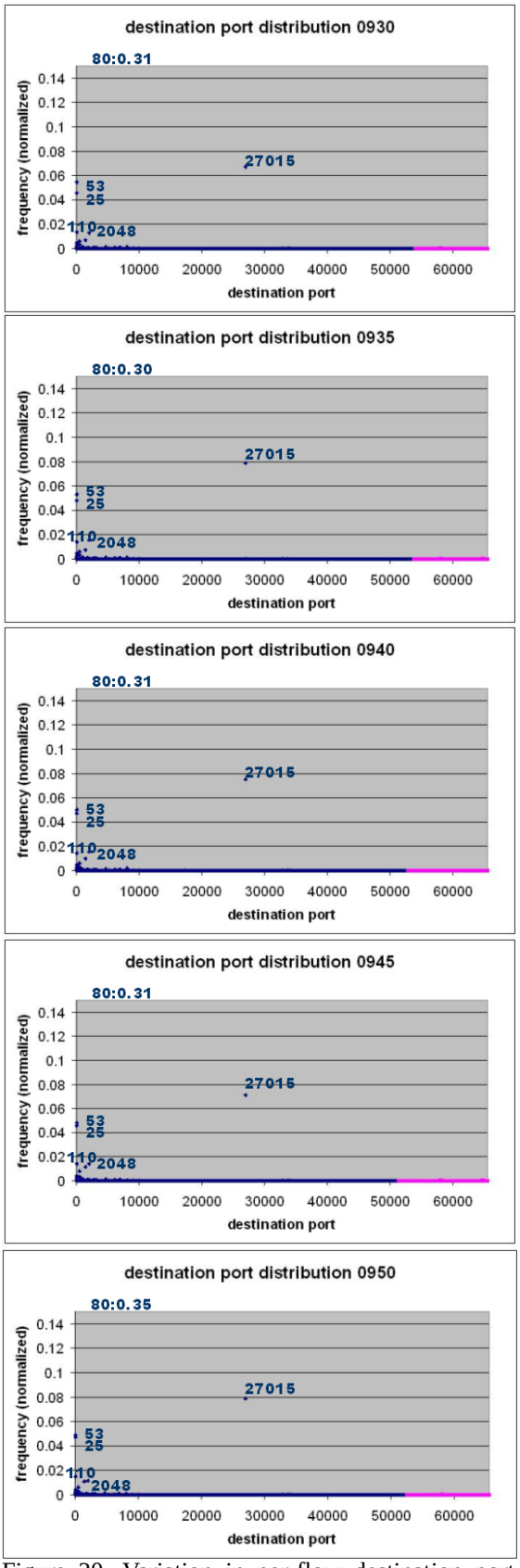


Figure 20. Variation in per-flow destination port distribution in 0905-0955 (continued).

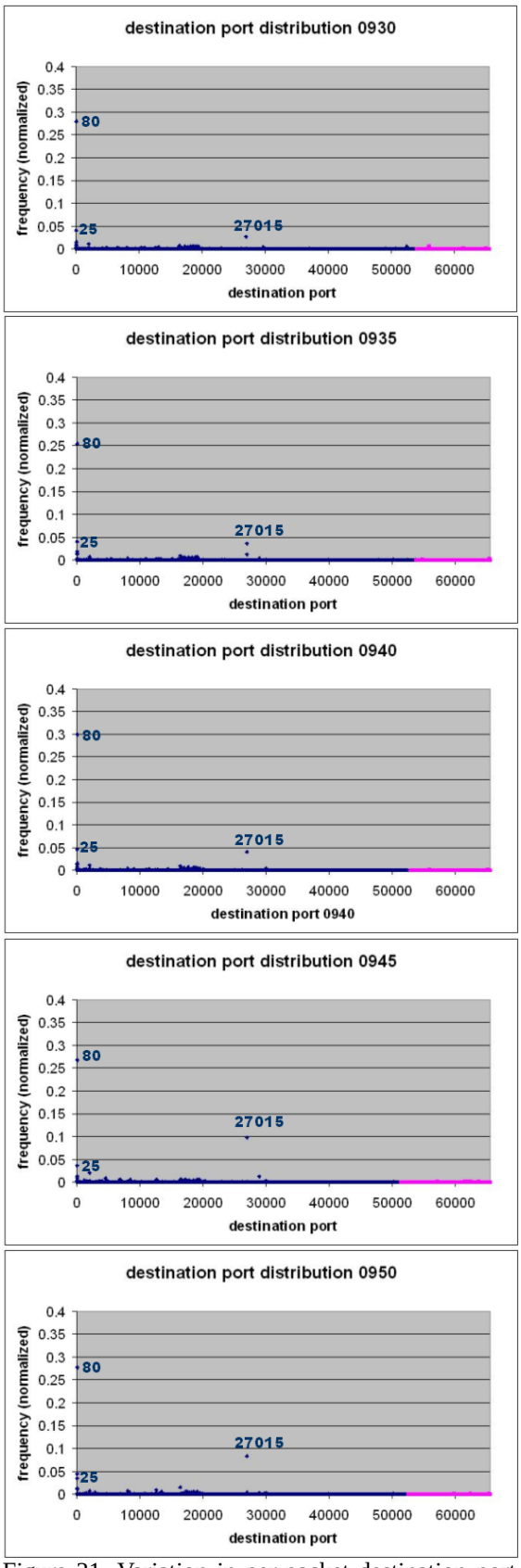


Figure 21. Variation in per-packet destination port distribution in 0905-0955. (continued).

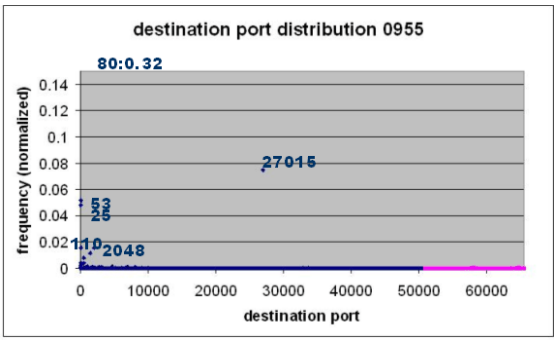


Figure 22. Variation in per-flow destination port distribution in 0905-0955 (continued).

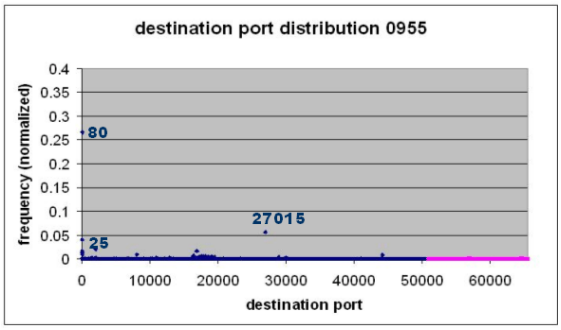


Figure 23. Variation in per-packet source port distribution in 0905-0955 (continued).

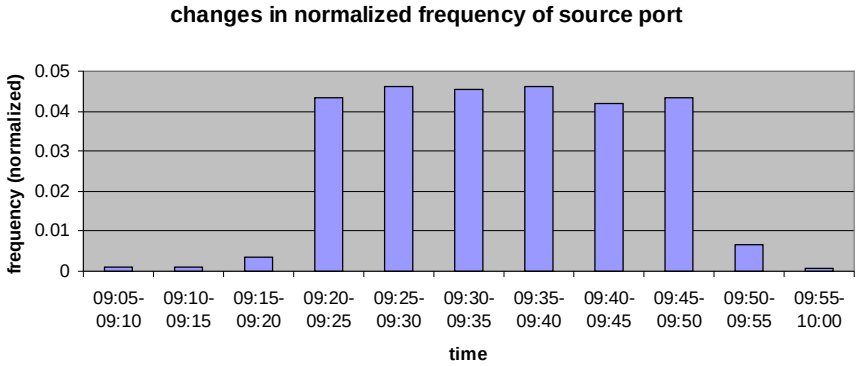


Figure 24. Changes in normalized frequency of source port in 5-minute interval on per-flow basis.

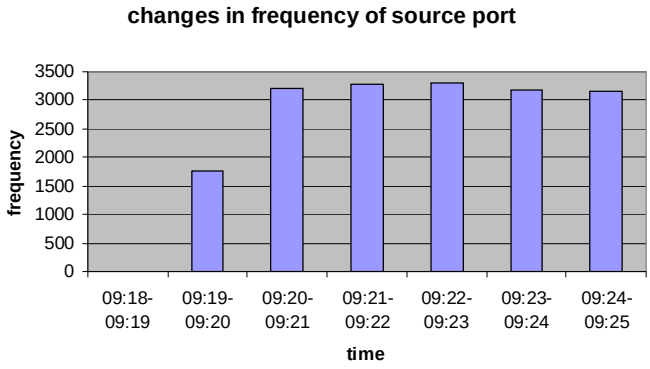


Figure 25. Changes in frequency of the attack packet signature in 1-minute interval on per-flow basis.

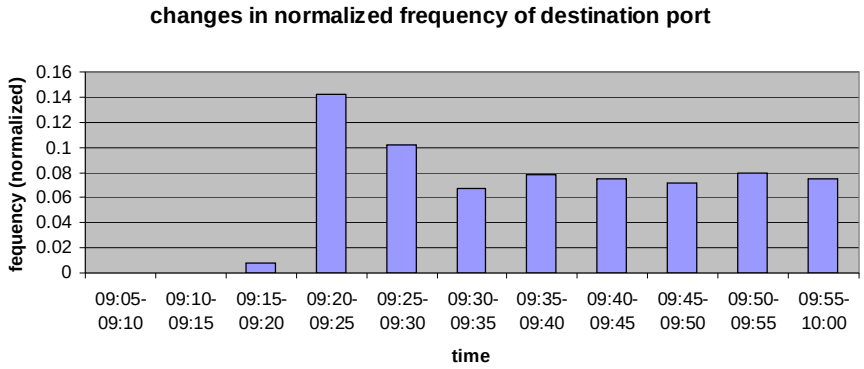


Figure 26. Changes in normalized frequency of destination port in 5-minute interval on per-flow basis.

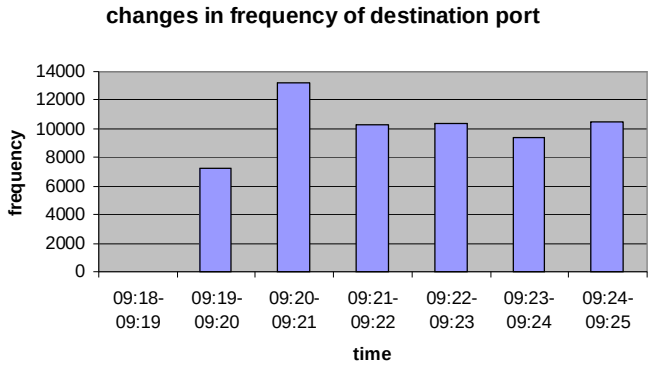


Figure 27. Changes in frequency of the attack packet signature in 1-minute interval on per-flow basis.

From the above observation, if attacks can be identified within two minutes, i.e., before the attack packets occupy a significant portion of the traffic, fewer packets will be received by the victim. Therefore, less resource will be used to handle requests originated from the attack packets, and the damages to the victim can be reduced.

3.2 Multiple SYN packets from one source

In the data provided by iAdvantage, there are 24,178,710 flow records observed, in which 972482 flows are originated from or destined to the victim host. 871,123 flow records are in forward direction (destined to the victim), while only 101,359 flow record are in reverse direction (originated from the victim). This illustrated the highly asymmetric nature of the traffic pattern generated by the DDoS attacks.

Although there are 871,123 flow records in forward direction, they are originated from 54 zombies only. The following snapshot of statistics shows that one source sent out a large amount of SYN packets in a DDoS attack.

```
IP               # SYN packets
200.48.156.160 : 8775
202.166.69.17  : 4775
202.56.197.137 : 184133
203.118.170.230: 62328
203.75.22.162  : 4210
210.193.249.194: 26565
211.192.243.189: 96
211.192.243.206: 163
211.197.57.244 : 486
211.21.119.98  : 9721
```

Figure 28. Multiple SYN packets are generated from one source.

Besides, a source can utilize multiple ports to send out the large amount of SYN packets. The following snapshot of statistics shows that multiple ports are used by a single source to generate an attack. Analysis also shows that the port used in this particular attack lies in the range between 1025 and 5000, and the distribution of port usage follows a nearly uniform distribution.

```
IP              ,port: # SYN packet
203.118.170.230,2001: 20
203.118.170.230,2002: 20
203.118.170.230,2003: 14
203.118.170.230,2004: 22
203.118.170.230,2005: 18
203.118.170.230,2006: 20
203.118.170.230,2007: 19
203.118.170.230,2008: 19
```

Figure 29.Uniform usage of ports in generating SYN packets.

To filter out such kind of attack, three options may be used:

- Filtering out the packets destined to the victim, according to destination IP and destination port.
- Locating the hosts sending multiple SYN packets through multiple ports to the victim, and filtering the packets according to the destination IP, destination port and source IP.
- Locate the packets that has the signature same as the identified attack packets, and filter these packets accordingly.

The first method can filter out all the packets destined to the victim. However, this will prevent the victim from hosting services, since no host can connect to the victim through the port utilized in the attack. So the probability of discarding normal packets may be too large to be acceptable.

The other two methods do not have this problem. However, the third method needs to check all the fields to draw conclusion on whether the packets received are legitimate or not. This may consume too much time and does not fulfil the need of reducing the time to detect and filter the attack packets. As a result, the second method is the suitable against such kind of attacks. Moreover, the number of access lists to filter the packets generated by the second method is significantly less than

that generated by the third one, this also illustrates that the second method uses much less time in detecting and filtering the attack packets.

3.3 Different forward and reverse paths of attack packets

From the data in iAdvantage network, we found that in the forward direction, there are 54 zombies. In the reverse direction, there are 122 zombies. There are only 43 zombies appear in both forward and reverse direction. This shows that the paths for attack packets in forward and reverse direction need not be the same path. This is a direct consequence that the current Internet is a packet-switching network.

For the routers close to the victim, e.g., the router directly connected to the victim, they may obtain the full set of attacking hosts, and therefore these routers may filter out all the attack packets. However, the routers farther away from the victim, like backbone routers in ISP, may not have the full set of attacking hosts since the attack packets can go through different paths to the victim. If all the routers detect and filter the attack packets independently, the same action may be repeated in different routers to obtain the same set of attacking hosts. This wastes resources in routers. Also, when the attack packets change their path, the whole process will need to be repeated in routers. If this cannot be done within a short time, the filtering may not be effective, and as a result, the filtering task will be shifted to the routers close to the victim, which may overload those routers.

Therefore it would be desirable to have a protocol between routers to exchange information about the detected intrusion. If such a protocol exists, the backbone routers can co-operate to identify the full set of the attacking hosts. When the attack packets change their paths, since the full set of attacking hosts are already propagated to all the backbone routers, these packets can be filtered without wasting time to detect. Also, the detection task can be distributed among all the backbone routers, so that the workload can be shared equally among the routers. And resources can be saved since routers can co-operate to get the full set of attacking hosts.

4. Summary

We observe that the histograms of packet attributes in normal condition and in attacking condition are quite different. The changes in the normalized frequencies of packet attributes due to attack is greater than the fluctuation of the normalized frequencies of packet attributes in normal condition. To detect possible attack, we may set a certain threshold which indicates the maximal allowed fluctuation in normalized frequencies. If the fluctuation is greater than the threshold, we may declare there is an attack.

Time is needed for an attack to generate enough traffic. So, another way to detect attack traffic is to monitor the histograms and see if there are any abnormal, sharp rises in the normalized frequencies of packet attributes. If such packet attribute exists, the packets having such attribute value may be the attack packets.

We also observe that one host may send out multiple SYN packets through multiple ports to the victim. So, to filter such kind of attack, we can locate the victim and the attacking hosts, and filter according to destination IP, destination port and source IP.

Furthermore, a protocol between routers to exchange information about the detected intrusion may be desired. This is useful for the routers to co-operate to obtain the full set of attacking hosts.

Acknowledgement

- This work is sponsored by the Areas of Excellence scheme established under the University Grant Committee of the Hong Kong Special Administrative Region, China (Project number AoE/E-01/99).
- The Information Engineering Department data was provided by Alan Lam.
- The iAdvantage data was provided by Ben Li and Ringo Hung, who also generously shared with us their knowledge about cyber attacks.

Appendices

Remark 1.
All the NetFlow data files are preprocessed by flow-tools. [3] Being processed by flow-tools, additional information is obtained, e.g., number of flows lost, number of packets corrupted.

Remark 2.
Because NetFlow export uses UDP to send export datagrams, it is possible for datagrams to be lost. [4] In all the NetFlow data files, some flows are lost. In general, only about 0.1% of the flow records are lost. So the lost flows have insignificant effect on the result.

Remark 3.
The flow records would not be exported if the end of flow is not detected. So if there are flows across two or more intervals, the flow records would be considered solely belonging to the interval that the flow ends. This may cause overestimation of the frequency of packet attributes in one interval, and underestimation in other intervals.

A flow can be exported under the following conditions: [5]

- Flows which have been idle for a specified time are expired and removed from the cache.
- Long-lived flows are expired and removed from the cache (flows are not allowed to live more than 30 minutes by default, the underlying packet conversation remains undisturbed).
- As the cache becomes full, a number of heuristics are applied to aggressively age groups of flows simultaneously.
- TCP connections which have reached the end of byte stream (FIN) or which have been reset (RST).

Remark 4.
The average packet length is calculated by dividing the byte count per flow by the packet count per flow. Ideally, we would use the byte count per individual packet to build the histogram. However, the records in NetFlow database are saved on a per-flow basis, not on a per-packet basis. So the average packet length was used instead.

Reference

[1] NetFlow Services Solutions Guide:
http://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/netflsol/nfwhite.htm
[2] Inside the Slammer Worm: http://www.computer.org/security/v1n4/j4wea.htm
[3] Flow tools: http://www.splintered.net/sw/flow-tools/
[4] NetFlow Export Datagram Format:
http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/nfc/nfc_3_0/nfc_ug/nfcform.htm
[5] White Paper NetFlow Services and Applications:
http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neflct/tech/napps_wp.htm
[6] MRTG: The Multi Router Traffic Grapher: http://people.ee.ethz.ch/~oetiker/webtools/mrtg/
[7] zlib: http://www.gzip.org/zlib/
[8] Steve Gibson, *DRDoS: Distributed Reflection Denial of Service, Description and analysis of a potent, increasingly prevalent, and worrisome Internet attack.* Gibson Research Corporation. February 2002.