Cheng and Tang *Journal of Internet Services and Applications* 2013, **4**:9
http://www.jisajournal.com/content/4/1/9

Journal of Internet Services and Applications
a SpringerOpen Journal

RESEARCH Open Access

PortView: identifying port roles based on port fuzzy macroscopic behavior

Guang Cheng[1*] and Yongning Tang[2]

Abstract

Port is a basic parameter in TCP/IP data communication. However, a port by its number has no clear association to the types of applications. Identifying such an association, which is referred to as a port role, provides important information to many network applications such as IDS and traffic shaper, as well as detecting new network services and security attacks. Traditional studies on port role identification are only based on port behavior shown on an individual host, other than jointly viewed macroscopic port behavior embodied by all relevant traffic flows among multiple hosts. Port role identification based on macroscopic behavior can reflect severs or clients to discover new services or attacks in the network. In this paper, we propose a novel port role identification approach called PortView, which is based on fuzzy macroscopic port behavior analysis. In our approach, we design a two-dimensional Macroscopic Port Classification Plane (MPCP) to classify port roles into six role zones using an EM fuzzy clustering algorithm. Comprehensive experiments using real Internet traces from the CERNET network validate that PortView can effectively and accurately identify port role based on its macroscopic port behavior.

Keywords: Port role identification, Macroscopic behavior

1. Introduction

Most current Internet applications are based on TCP or UDP. Traditionally, network applications use fixed port numbers assigned by IANA [1]. Accordingly, TCP and UDP port numbers are directly associated to the types of various network application, e.g. TCP/80 for HTTP etc. TCP and UDP ports identified with 16-bit numbers can be partitioned into three types: (1) Well-known Ports, which are port numbers between 0 to 1023, and closely bound to system specific services and protocols; (2) Registered Ports, which are port numbers between 1024 to 49151, and used by processes of procedures and executed programs of common users; and (3) Private Ports, which are port numbers between 49152 to 65535, and dynamically or temporarily used by client communication processes or executed programs.

Each TCP or UDP port used in a TCP/IP network application implies a communication channel, which can also be exploited by an intruder. Port exploitation based network intrusion are mainly manifested in the following two styles: (1) exploiting common ports, and attacking hosts through known system bugs; (2) planting Trojan horses to victim hosts through backdoor opened by other network vulnerabilities. It is worth noting that port numbers used by various malicious software (e.g., Trojan programs) are widely distributed, which don't only use registered ports. Port scanning is commonly used by an intruder to search for potential targets. Network administrators can monitor their networks by tracking abnormal port roles, such as a client host provides services outwards the public. Many other port related behaviors such as the changes of port in/out degreecan help check whether or not a host behaves normally. In general, port profile is a very important metric for application traffic classification. Correctly identifying server ports among network traffic has great significance for accurately classifying traffic application types.

At present, few studies focus on identifying port roles, i.e., a port belongs to a server or client port. Many metwork security and management applications, such as port scanning, worm detecting, traffic classification etc., are closely related to port role identification. In this paper, we define and analyze port roles based on their macroscopic traffic behavior. Macroscopic port roles

* Correspondence: gcheng@njnet.edu.cn
[1]School of Computer Science and Engineering, Key Laboratory of Computer Network and Information Integration, MOE, Southeast University, Nanjing, P.R. China
Full list of author information is available at the end of the article

first can be classified into active ports and non-active ports, and then active ports are classified into normal behavior roles and abnormal behavior roles. Abnormal behavior roles are classified into scan attacked ports and service failed ports, and normal behavior ports can be classified into client ports, server ports, or p2p ports etc. For this purpose, we use data mining to classify port roles from massive traffic data automatically. The contributions in this paper can be summarized as the following:

- Define port role to distinguish different communication pattern.
- Propose a two-dimensional Macroscopic Port Classification Plane (MPCP) that creates a new space for traffic behavior analysis.
- Propose an EM fuzzy clustering algorithm to classify port traffic into six behavior zones in MPCP.

The rest of the paper is organized as follows. Firstly, we analyze Macroscopic Port Role in Section 2. We design a two dimensional Macroscopic Port Classification Plane and propose an EM clustering algorithm for fuzzy port behavior analysis in Section 3. Extensive experiments using real Internet traces to validate PortView are shown in Section 4. The related work is presented in Section 5. Finally Section 6 concludes this paper.

2. Macroscopic port role analysis

In this section, we will introduce the definition of Macroscopic Port Role as a new defining traffic characteristic, and use it re-profile several common network applications or behaviors.

2.1 Macroscopic port role

With the intent to define Macroscopic Port Role, we first define several port related parameters.

Let SF_i be a set of flows with source port i such that $\forall f \in SF_i$, *getSrcPort*(f) = i. We denote n_i=|SF_i| as the size of SF_i. Here, *getSrcPort*(f) is the source port number of the flow f, retrieved via the function *getSrcPort*(). Similarly, *getDstPort*(), *getSrcIP*() and *getDstIP*() are the functions to retrieve destination port, source IP and destination IP from the flow f. Accordingly, a flow f is uniquely defined as f = {*getSrcIP*(f), *getDstIP*(f), *getSrcPort*(f), *getDstPort*(f)}.

All distinct source IP addresses retrieved from the flows in SF_i, which are related to the source port i, constitutes the source IP set denoted as $SSIP_i$. The number of all distinct source IP addresses with source port i is denoted as |$SSIP_{i\cdot}$|. Similarly, all distinct destination IP addresses retrieved from the flows in SF_i constitute the destination IP set denoted as $SDIP_i$. The number of all distinct source IP addresses with destination port i is denoted as |$SDIP_{i\cdot}$|.

Let DF_j be a set of flows such that $\forall f \in DF_j$, *getDstPort* (f) = j. We denote m_j=|DF_j| as the size of DF_j. All distinct source IP addresses retrieved from the flows in DF_j, which are related to the destination port j, constitute the source IP set denoted as $DSIP_j$. The number of all distinct source IP addresses with destination port i is denoted as |$DSIP_i$|. .Similarly, all distinct destination IP addresses retrieved from the flows in DF_j constitutes the destination IP set denoted as $DDIP_j$. The number of all distinct destination IP addresses with destination port i is denoted as |$DDIP_i$|.

Definition 1. For a given port i, its *Macroscopic Port Role* (MPR) observed on a monitored network link during a measurement period T is defined by the 4-tuple port parameters < $SSIP_i$, $SDIP_i$, $DDIP_i$, $DSIP_i$ >.

For a source (or destination) port i, if the corresponding $SSIP_i$ and $SDIP_i$ (or $DSIP_i$ and $DDIP_i$) are larger than a threshold, it is called active port. Otherwise, port i is called inactive port. The threshold will be computed by the Port Role Interval Classifying Algorithm in Section 3.4.

In the following, we broadly classify network traffic based on its role into two categories: abnormal and normal network roles. Then, for each category, we re-profile several selected common network applications or network scenarios using MPR analysis to illustrate PortView. It is worth noting that the classification criteria shown in this paper is not the only way to use PortView. The more granular classification criteria are, the more zones we may create in MPCP as discussed later in Section 3.

The different port role of MPR is described from Figures 1, 2, 3, 4, 5. In these figures, the left side of the dotted line means that the DIP hosts send some

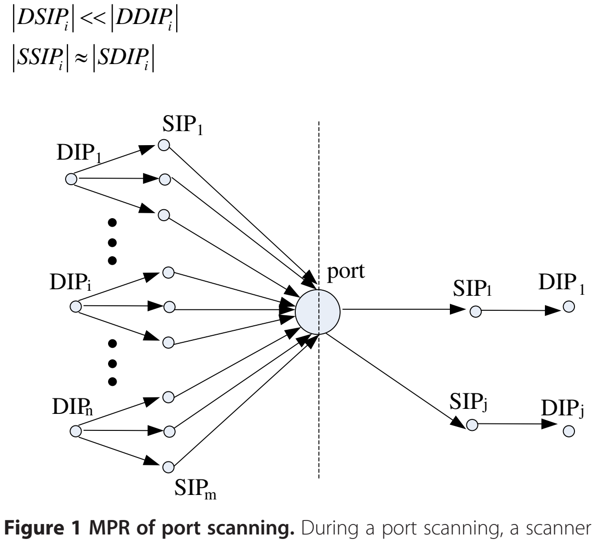

Figure 1 MPR of port scanning. During a port scanning, a scanner attempts to connect to a specific port of a large number of hosts.

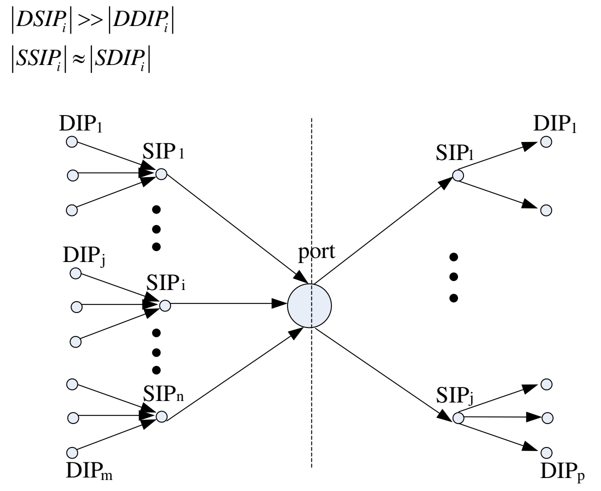

Figure 2 MPR of failed service. In a typical DDoS attack scenario, a large number of DIPs send requests connecting to a specific port i on a few SIPs with destination port i.

connection requests to an port *i* on the *SIP* hosts. The right side of the dotted line means that *SIP* hosts with port *i* send some connection requests to the *SIP* hosts. These figures show the Macroscopic Port Role.

2.2 MPR of abnormal traffic

In the traffic category of abnormal network role, we study the port role of (1) commonly observed port scanning traffic, and (2) a network service under DDoS attack.

1) *MPR of port scanning*: The MPR of port scanning can be illustrated in Figure 1, which clearly shows that on the left side of the dotted line, a few hosts denoted as *DIP* (Destination IP address) send too many connection requests to an interested port *i* possibly open on too many hosts denoted as *SIP* (Source IP address) with destination port *i*. Thus, we have |*DSIP_i*|<<|*DDIP_i*|. Usually only a few *SIPs* (Source IP address) with this port *i* open response to the connection requests with source port *i*, as shown on the right side of the dotted line in Figure 1, which can be described with: |*SSIP_i*| ≈ |*SDIP_i*|. Hence, we have the following MPR of port scanning:

$$\begin{aligned}|DSIP_i| &<< |DDIP_i|\\ |SSIP_i| &\approx |SDIP_i|\end{aligned} \tag{1}$$

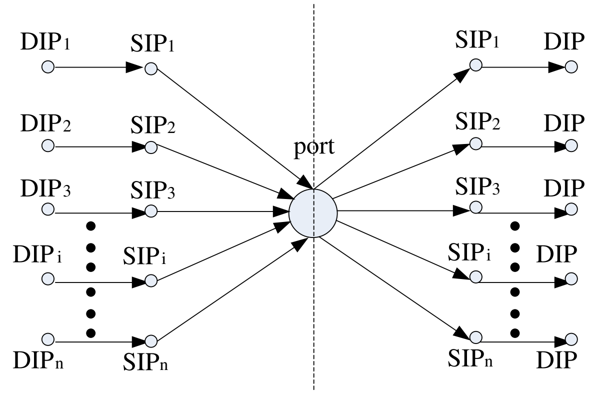

Figure 3 MPR of client application. The port on the client is usually chosen by the operating system dynamically. Different ports are used to communicate with multiple servers.

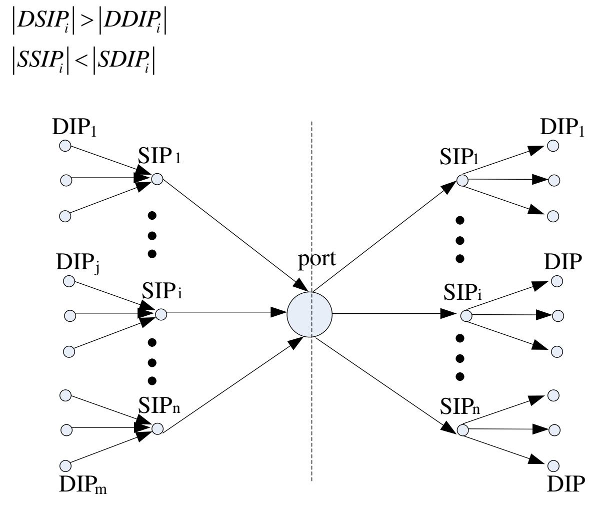

Figure 4 MPR of server application. A service port is the one used by a server listening to the requests from different clients, so the number of servers is usually less than the number of clients.

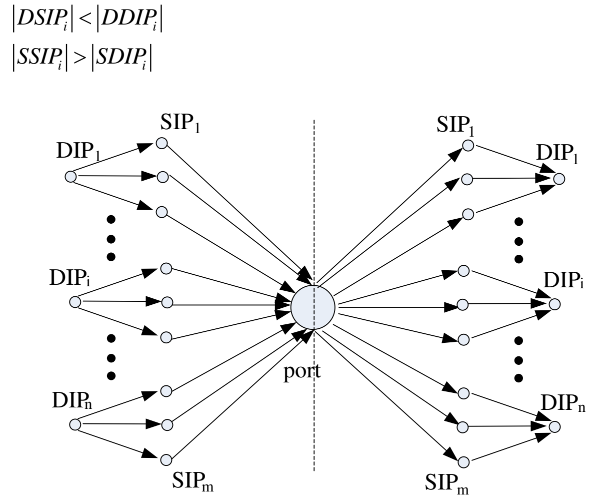

Figure 5 MPR in overlay application. Typically, an overlay node firstly obtains a list of neighbor nodes, and then sends request to each selected node. Hence, in an overlay network which uses a unified service port number to provide service, the communication pattern is that few clients send requests to lots of hosts.

2) *MPR of failed network service:* In a typical DDoS attack scenario, a large number of *DIPs* send requests connecting to a specific port *i* on a few *SIPs* with destination port *i*, that implies $|DSIP_i|>>|DDIP_i|$. For those hosts being DDoS attacked severely, they cannot response interactively. Thus, only a few *SIPs* as shown on the right side of the dotted line in Figure 2 can response to the connection requests with source port *i*, which implies: $|SSIP_i| \approx |SDIP_i|$. Thus, the MPR of failed service caused by successful DDoS attack can be described as:

$$\begin{aligned} &|DSIP_i| >> |DDIP_i| \\ &|SSIP_i| \approx |SDIP_i| \end{aligned} \tag{2}$$

2.3 MPR of normal port behaviors

In the traffic category of normal network behavior, we study the port behavior of (1) client application, (2) server application, and (3) overlay application.

1) *MPR of client application:* In a typical client–server network application, for every monitored network flow $sf_i = (ip_1, ip_2, i, j)$ destined to the host ip_2, a corresponding opposite directional flow $df_i = (ip_2, ip_1, j, i)$ should appear. We denote such a pair of flows < sf_i , df_i > as <in-flow, outflow> pair. In this paper, we denote the MPR of client–server applications as regular port behavior, which can be defined as:

$$\begin{aligned} &|SSIP_i| = |DDIP_i| \\ &|SDIP_i| = |DSIP_i| \end{aligned} \tag{3}$$

In the most common client–server paradigm (C-S mode), a client with IP address srcIP communicates with a server with IP address dstIP. The port on the client is usually chosen by the client's operating system dynamically. Different ports are used if the same client needs to communicate with multiple servers. However, different clients may happen to choose the same port to communicate to different servers. We show the port behavior in C-S mode in Figure 3, where *SIP* and *DIP* represent client and server hosts respectively. As shown on the left of the dotted line in Figure 4, different SIP addresses with the same port communicate with DIP addresses, while DIP addresses response accordingly. We can see that a given port links a *DIP* to a *SIP*, and different SIP addresses may use the same port to communicate to their connecting DIP addresses. Clearly, the number of SIP addresses is equal to the number of DIP addresses. Thus, we have $|DSIP_i| = |DDIP_i|$. On the right side of the dotted line, we see that the number of SIP addresses is equal to the number of DIP addresses. Thus, we have $|SSIP_i| = |SDIP_i|$. It is possible that some connecting ports from source IP addresses have not received response from the destination IP addresses. However, even if in such a case, the number of SIP addresses is still equal to the number of DIP addresses since a source IP and a destination IP always emerge in pair in a network flow.

There are further two possible port behavior variations in C-S mode: (1) Different source IP addresses use the same port to communicate to a single destination IP. In this case, the number of source IP addresses is larger than the number of destination IP addresses, namely, $|SSIP_i| \geq |SDIP_i|$. (2) A few source IP addresses send information frequently during the measurement interval. The source port used by a source IP reverses, at the same time, the other source IP uses the same port number to access a different destination IP, the destination IPs which the two source IPs access is random, From the probability theory standpoint, the number of source IPs and destination IPs in flow records in which the client port takes role as a source port are almost equal, namely, $|DSIP_i| \approx |DDIP_i|$. Hence, we have the following observation on the port behavior of C-S mode:

$$\begin{aligned} &|DSIP_i| \approx |DDIP_i| \\ &|SSIP_i| \approx |SDIP_i| \end{aligned} \tag{4}$$

2) *MPR of server application:* A service port is the one used by a server listening to the requests from different clients. Obviously, the number of servers is usually less than the number of clients. The port behavior of a service port displays that multiple clients marked by *DIP* may send requests to a service port of the servers SIP addresses as shown on the left side of the dotted line in Figure 4. The right side of the dotted line in Figure 4 shows the servers response accordingly from this service port of SIP addresses. Hence, for a service port, there are more srcIPs than dstIPs in its in-flow direction, i.e., $|DSIP_i| > |DDIP_i|$. Since each SIP should answer several clients' requests, the number of srcIPs is less than that of dstIPs in their out-flow direction, i.e., $|SSIP_i| < |SDIP_i|$. Hence, we have the following observation on the port behavior of service port:

$$\begin{aligned} &|DSIP_i| > |DDIP_i| \\ &|SSIP_i| < |SDIP_i| \end{aligned} \tag{5}$$

In traditional C-S model, a server may have multiple connections to clients. However, a P2P host is limited to provide only one connection to each P2P. Suppose the number of links a server provided is fixed, a P2P host can provide service for more P2P. Hence, we have the following observation on the port behavior of a P2P host:

$$\begin{aligned}&|DSIP_i| >> |DDIP_i|\\&|SSIP_i| << |SDIP_i|\end{aligned} \tag{6}$$

3) *MPR of overlay application:* In an overlay network, a node uses a unified service port to provide service to other nodes. Typically, an overlay node firstly obtains a list of neighbor nodes, and then sends request to each selected node. Hence, in an overlay network which uses a unified service port number to provide service, the communication pattern is that few clients send requests to lots of hosts. Figure 5 shows the communication pattern based on a unified port of an overlay network. On the left side of the dotted line of Figure 5, few clients send requests to lots of hosts, i.e., $|DSIP_i|<|DDIP_i|$; On the right side of the dotted line, lots of hosts answer the requests to few clients, i.e., $|SSIP_i|>|SDIP_i|$. Hence, we have the following observation on the port behavior of a unified port in an overlay network:

$$\begin{aligned}&|DSIP_i| < |DDIP_i|\\&|SSIP_i| > |SDIP_i|\end{aligned} \tag{7}$$

3. Classifying port role algorithm

3.1 Port role metrics

As discussed in Sec. 2, the Macroscopic Port Role analysis can characterize communication patterns as shown in Eq. 1~Eq. 7. For facilitating MPR based traffic classification, we design a Macroscopic Port Classification Plane (MPCP) with X-Y axes representing two different behavior metrics. Then, we propose a fuzzy port behavior analysis method to divide MPCP into multiple zones with each zone representing different network behavior and corresponding network applications.

Definition 2. Port Exporting Network Activity X_i: for all network flows leaving from port i in a measurement time window, the ratio of the difference between the numbers of related source and destination hosts and the number of all the port i related hosts that send traffic is called port i's Port Exporting Network Activity X_i.

$$X_i = \frac{|SDIP_i| - |SSIP_i|}{|SDIP_i| + |SSIP_i|} \tag{8}$$

Definition 3. Port Importing Network Activity Y_i: for all network flows destining to port i in a measurement time window, the ratio of the difference between the numbers of related source and destination hosts and the number of all the port i related hosts is called port i's Port Importing Network Activity Y_i.

$$Y_i = \frac{|DSIP_i| - |DDIP_i|}{|DSIP_i| + |DDIP_i|} \tag{9}$$

From Definition 2 and 3, $|SDIP_i|-|SSIP_i|<|SDIP_i|+|SSIP_i|$, $|DSIP_i|-|DDIP_i| < |DDIP_i|+|DDIP_i|$. Thus, we have $-1 < X_i, Y_i < 1, i \in [0, 65535]$.

Definition 4. Port Differential Network Activity B_i: for a given port i, the difference between its Port Exporting Network Activity X_i and Port Importing Network Activity Y_i.

$$B_i = X_i - Y_i = \frac{|SDIP_i| - |SSIP_i|}{|SDIP_i| + |SSIP_i|} - \frac{|DSIP_i| - |DDIP_i|}{|DSIP_i| + |DDIP_i|} \tag{10}$$

Definition 5. Port Accumulative Network Activity A_i: for a given port i, the accumulation of its Port Exporting Network Activity X_i and Port Importing Network Activity Y_i.

$$A_i = X_i + Y_i = \frac{|SDIP_i| - |SSIP_i|}{|SDIP_i| + |SSIP_i|} + \frac{|DSIP_i| - |DDIP_i|}{|DSIP_i| + |DDIP_i|} \tag{11}$$

Definition 6. Port Overall Network Activity D_i: for a given port i during a measurement time window, the whole numbers of port i's related source and destination hosts. In other word, it is the all number of hosts related to source or destination port i.

$$D_i = |SSIP_i| + |SDIP_i| + |DSIP_i| + \left|DDIP_i^{\,i}\right| \tag{12}$$

X_i and Y_i are the basic clustering metrics, which are used to classify port roles into six different zones. A and B are two derived parameters from X and Y, and used directly in the EM clustering algorithm to categorize active ports.

3.2 Fuzzy port behavior

The characteristics of port behavior can be summarized in Table 1. In the following, we introduce Macroscopic Port Classification Plane (MPCP). Then we propose fuzzy port behavior analysis to quantitatively map all earlier discussed network applications or scenarios into the corresponding MPCP zones.

The Macroscopic Port Classification Plane is a 2-dimensional graph, as shown in Figure 6, where X axis represents Port Exporting Network Activity, and Y axis displays Port Importing Network Activity. In the case of normal network behavior, the corresponding Port

Table 1 Port communication behavior

Port status	Basic metrics	Port role	Communication
Normal Port	$X_i \approx Y_i$	Client Port	$\lvert DSIP_i\rvert \approx \lvert DDIP_i\rvert$ $\lvert SSIP_i\rvert \approx \lvert SDIP_i\rvert$
		Service Port	$\lvert DSIP_i\rvert > \lvert DDIP_i\rvert$ $\lvert SSIP_i\rvert < \lvert SDIP_i\rvert$
		P2P Service Port	$\lvert DSIP_i\rvert >> \lvert DDIP_i\rvert$ $\lvert SSIP_i\rvert << \lvert SDIP_i\rvert$
		Overlay Port	$\lvert DSIP_i\rvert < \lvert DDIP_i\rvert$ $\lvert SSIP_i\rvert > \lvert SDIP_i\rvert$
Abnormal Port	$X_i \neq Y_i$	Scanning Port	$\lvert DSIP_i\rvert << \lvert DDIP_i\rvert$ $\lvert SSIP_i\rvert \approx \lvert SDIP_i\rvert$
		Failed Service Port	$\lvert DSIP_i\rvert >> \lvert DDIP_i\rvert$ $\lvert SSIP_i\rvert \approx \lvert SDIP_i\rvert$

Exporting and Importing Network Activities manifest $X_i \approx Y_i$ (B_i is close to 0) based on Eq. 3. Thus, if the MPR analysis result shows the traffic is spanned along the diagonal area of MPCP (marked as zones 1, 2, 3 and 4 in Figure 7), the corresponding traffic belongs to normal network behavior; otherwise abnormal and the MPR analysis results should show the traffic is either located in zone 5 or 6 in Figure 6. If $-\delta \le B_i \le \beta$, $\delta > 0$ and $\beta < 1$, it belongs to normal port behavior. As a failed service port, according to Eq. 1, its Port Exporting Network Activity is less than its Port Importing Network Activity, $X_i < Y_i$. It falls into zone 5. As $Bi < -\delta$, port i falls into zone 5.

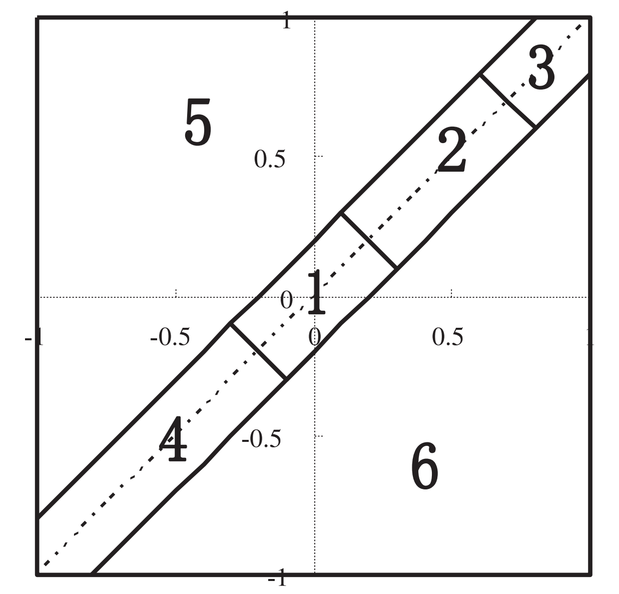

Figure 6 Macroscopic port classification plane. X axis represents Port Exporting Network Activity, and Y axis displays Port Importing Network Activity. The range of six zones can be showed in Table 2.

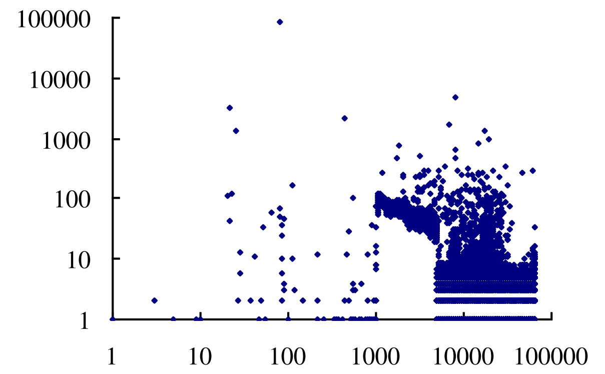

Figure 7 Distribution of flows. It shows the distribution of flow records with the changing of source port from 1 to 65535.

According to the analysis of Sec. 3.2 (1), in the case of abnormal network behavior, the corresponding Port Exporting and Importing Network Activities are different: $X_i \neq Y_i$. For scanned port, based on communication relationship as Eq. 1, its Port Exporting Network Activity is more than its Port Importing Network Activity, $X_i > Y_i$. Thus, it is in zone 6, and $B_i > \beta$.

For zone 1, 2, 3 and 4 in the diagonal area, in addition to the condition $-\delta < B_i < \beta$, according to the relationship of client port in Table 1, port i's Port Accumulative Network Activity $A_i \approx 0$, $-\phi \le A_i \le \varphi$, and $0 < \phi, \varphi < 1$. Overlay service port is in zone 4, $A_i < -\phi$. Service port is in zone 2, $A_i > \varphi$. For the boundary between zone 2 and 3, according to the classification of service port and p2p port in Table 1, in zone 2, $\varphi < A_i \le 2 - \gamma$, and in zone 3, $2 - \gamma < A_i < 2$. In conclusion, the range of six zones can be showed in Table 2.

Table 2 shows that in order to classify the 6 roles, we need to determine five arguments δ, β, ϕ, φ, and γ. In the following, we will design a clustering method EM algorithm that can classify the two kinds of distributions.

3.3 EM algorithm for Two classes of distribution

In this section, we will discuss how to determine the boundary arguments for each zone. Observing that each port may play certain role (i.e., projected into certain zone in MPCP) with some probability, we use statistical clustering to determine zone boundaries in MPCP. The basis of

Table 2 Port behavior zones

Zone ID	Port role	Port zones
1	Client Port	$-\delta \le B_i \le \beta, -\varphi \le A_i \le \varphi$
2	Service Port	$-\delta \le B_i \le \beta, \varphi < A_i \le 2-\gamma$
3	P2P Port	$-\delta \le B_i \le \beta, 2-\gamma \le A_i \le 2$
4	Overlay Port	$-\delta \le B_i \le \beta, A_i < -\varphi$
5	Scanning Port	$B_i < -\delta$
6	Failed Service Port	$B_i > \beta$

statistical clustering is to build a finite mixed statistical model such that it can use k probability distributions to replace k clustering. Since a port may belong to different zones/roles with different probabilities, we define a port role as the MPCP zone it is most likely located (i.e., with the biggest probability distribution).

For two kinds of statistical clustering, if we need to do this on A and B classes, every class is supposed to be normal distribution. Clustering A's expectation and variance are μ_A and σ_A. Clustering B's expectation and variance are μ_B and σ_B. We take samples from the distributions of A and B. The probability of taking A or B is p_A or p_B respectively. Here, p_A+p_B=1. Here, the challenge is how to determine five arguments, μ_A, σ_A, μ_B, σ_B, and p_A (or p_B), by only knowing the MPR analysis result.

Since the distribution that each port follows is unknown initially, we use the fuzzy clustering EM algorithm to classify the port behaviors. In every iterative step of classical k-average clustering algorithm, every port is considered exclusively belonging to a class. Thus, we adopt fuzzy clustering to tackle the challenge of multiple distributions a port may follow. Each port probabilistically belongs to a class. First, every port's class is estimated. Then, compute every port's clustering probability using initial estimated value. These probabilities are used to estimate the argument again. The above process is iteratively conducted for the second time. This method is called Expectation Maximization (EM). First, compute clustering probability, in another word, the expectation's class value; Second, Maximum likelihood the given data's distribution. The following test will show the EM algorithm in detail.

For n port's metric: $x_1, \cdots, x_n$, there are two classes A and B. z is a two-dimensional array. At the very beginning, let z_{Aj} be 1, meaning x_j is in A class; otherwise 0. Let z_{Bj} be 1, meaning x_j is in B class; otherwise 0. Then according to z's value, we can compute k class's argument using the following formula:

$$\mu_A = \frac{\sum_j E(z_{A_j}|x_j)\cdot x_j}{\sum_j E(z_{A_j}|x_j)}, \sigma_A = \frac{\sum_j E(z_{A_j}|x_j)\cdot (x_j-\mu_A)^2}{\sum_j E(z_{A_j}|x_j)}$$
$$\mu_B = \frac{\sum_j E(z_{B_j}|x_j)\cdot x_j}{\sum_j E(z_{B_j}|x_j)}, \sigma_B = \frac{\sum_j E(z_{B_j}|x_j)\cdot (x_j-\mu_B)^2}{\sum_j E(z_{B_j}|x_j)} \tag{13}$$

For a port i,

$$x_{1,2} = \frac{\mu_A\sigma_B^2 - \mu_B\sigma_A^2}{\sigma_A^2 - \sigma_B^2} \pm \frac{\sqrt{(\mu_A\sigma_B^2 - \mu_B\sigma_A^2)^2 + (\mu_A^2 - \mu_B^2)(\mu_A\sigma_B^2 - \mu_B\sigma_A^2 - 2\sigma_A^2\sigma_B^2 \ln(p_B\sigma_A/p_A\sigma_B))}}{\sigma_A^2 - \sigma_B^2} \tag{14}$$

The two class EM classifying method discussed in Eq. (14) can put data into 2 or 3 classes. As in Section 3.1, we provided several metrics that can classify port behaviors, and each time the method uses one dimension metric to do classification. The next subsection will give a detailed port role interval classifying algorithm based on the Eq. (14).

3.4 Port role interval classifying algorithm

In the measuring progress, classifying algorithm first records every port's source IP numbers, source port's destination IP numbers, destination port's source IP numbers, destination port's source IP numbers. After a measuring time period, the role style of the every port's four metric is determined. In measuring data, if the number of a port's related export and import hosts is small, then because of the tiny random difference between the numbers of source and destination hosts, some inactive port will be made with big random difference between port's inward and outward communication metric. At the same time, we care about these active ports with large traffic of the host that influence the network a lot. And we remove these ports with small traffic and have little influence on the network. In order to determine a port is an active port or not, we compute every port's activity metric D. According to the two-class EM algorithm in section 3.3, all the ports are classified into active and inactive ports. Because D is x_1, based on formula (19), we can compute active port's threshold. For all the ports whose activity metric value D_i is bigger than or equal to x_2, they are defined as active ports. Otherwise, they are inactive ports.

For all the active ports, using port's communication difference balance metric B, based on the two-class EM algorithm in section 3.3, we can compute the classification threshold of the normal and the abnormal port. The two-class EM algorithm's two distribution's interval can figure out normal port's range is $[x_1, x_2]$, and abnormal port's range is $(-\infty, x_1), (x_2, +\infty)$. Because port's communication difference balance metric B's range value is (−2, 2), abnormal port's value range is (−2, x_1), (x_2, 2). So we can compute the argument α, β, $\alpha = -x_1, \beta = x_2$ in Table 2.

For all the normal port, using port communication difference accumulation metric A, based on the two-class EM algorithm in section 3.3, we can compute the classification threshold of client port and service port. The two-class EM algorithm's two distribution's interval can figure out client port's range is $[x_1, x_2]$, and service port's range is $(-\infty, x_1)$, $(x_2, +\infty)$. Because *A*'s value range is $(-2, x_1)$, $(x_2, 2)$, service port's range is $(-2, x_1)$, $(x_2, 2)$. So we can compute the argument $\phi, \varphi, \phi = -x_1, \varphi = x_2$ in Table 2.

Service ports are divided into there kinds: overlay service port, normal service port and P2P service port. Using communication difference accumulation metric A, besides classifying the styles of client port, it can also classify overlay service port. So if it is a overlay port, A's value range is $(-2, -\phi)$. For the range $(\varphi, 2)$, it contains normal and P2P service port. We again use the two-class EM algorithm to classify these two kinds of ports. The result is that service port's range is (φ, x_1), and P2P's service port's value range is $(x_2, 2)$. So we get the argument $\gamma = 2 - x_2$. Based on the method above, we can compute the five arguments in Table 2. So we can classify all the active ports into the six kinds of port roles.

4. Experimental analysis

We will divide port roles as a whole. First, we introduce the experimental traces which are used for interval dividing of port roles. Second, we classify the ports into active ports and inactive ports according to the steps of algorithm in Section 3.4. For active ports, we classify it into normal ports and abnormal ports, and then we classify normal ports into four types of roles: client ports, server ports, P2P ports and overlay ports.

4.1 Experimental traces

Multiple real network traces were selected from the link between CERNET backbone and Jiangsu CERNET network to evaluate PortView. Specifically, we present two representational network traces in this section, namely Trace1 and Trace2. Trace1 was collected on April 10, 2010 with 175 million packets, 117 GB traffic traces, 495,000 flows, and its consecutive observation period is ten minutes. Trace2 was captured on May 4, 2012 with 2.82 million packets, 1.89 GB traffic traces, 17,000 flows, and its observation period is five second. Trace1 is selected to show how PortView performs given relatively rich correlation information among multiple hosts over a longer observation window (10 m). Trace2 is used to validate the agility of PortView given a short observation window (i.e., 5 s). Here, we define PortView agility as its ability to provide port role analysis result over a short time period (e.g., 5 s). Network flows are defined using 4-tuple, i.e. source address, destination address, source port, and destination port in this paper. Trace1 is analyzed from Subsection 4.1 to Subsection 4.3, and the analysis based on Trace2 will be reported in Subsection 4.4.

We can see clearly from Figure 7 that: (1) the number of flows with source port between 1 and 1023 is unevenly distributed; (2) the number of flows with port between 1024 and 5000 is distributed quite uniformly, every source port produces corresponding host amount, the number of host is between 20 and 100; and (3) the flows with port above 5000 is also unevenly distributed.

Figure 8 shows the two-dimensional distribution diagram of port communication difference. We can know from Figure 8 that ports distribute mainly near the origin. According to the Section 3, this type of ports are mainly client ports; In addition, there are a lot of ports which distribute near the diagonal of first quadrant, those ports are server ports, this indicates that most ports belong to normal port. Points in the negative diagonal of third quadrant indicate that the number of IP requesting to this port IP are less than requested port IP amount, this character is similar to scanning, but most of the scanning traffic has no response, that is to say the X-axis coordinate of scanning traffic is close to zero. Because of being in the inner of an overlay, most of requesting traffic has response from a server port of the overlay. There are few points in the third quadrant whose diagonal is near the X-axis, the character is similar to the points which are near the diagonal. Because the requesting traffic is less than the response traffic, points in the fourth quadrant indicate that parts of the traffic of this port are scanning attacks, such as 8088, 7155 and so on.

4.2 Port clustering

According to the activity metric value D of every port, two-class EM algorithm can cluster all 65536 ports into active and inactive ports. The corresponding five metrics $(\mu_A, \sigma_A, \mu_B, \sigma_B, p_A)$ = (162.426, 720.223, 9.297, 5.789, 0.1).

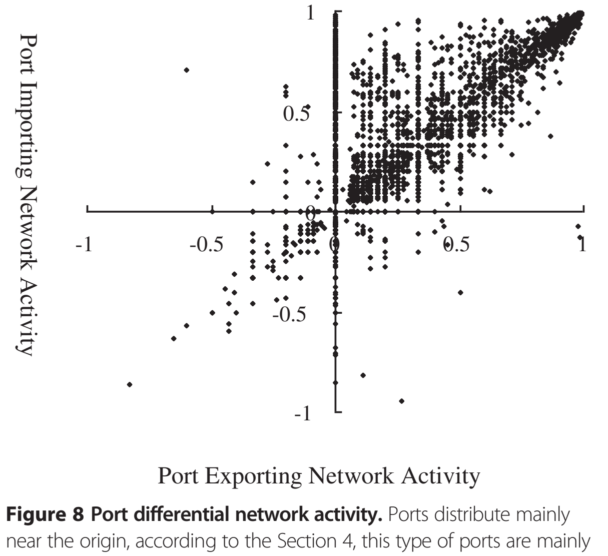

Figure 8 Port differential network activity. Ports distribute mainly near the origin, according to the Section 4, this type of ports are mainly client ports.

Thus we can calculate activity metric value, and classify those ports whose activity value no more than 30 into inactive port, cluster the port whose activity value more than 30 into active port. The number of active ports in this experiment is 5313. In the following, we will analyze 5313 clustered active ports. Figure 9 shows the two-dimensional graph of metric of host difference of 5313 active ports.

We calculate communication difference balance value B of all 5313 active ports, and use two-class EM algorithm to calculate two distribution parameter values (μ_A, σ_A, μ_B, σ_B, p_A) = (−0.2323, 0.3533, 0.0017, 0.084, 0.17). Hence, we can confirm the range of normal ports is [−0.0547, 0.0596], the range of abnormal ports is (−2,−0.0547) and (0.0596, 2). Correspondingly, we can calculate the parameters δ and β in Table 2 as δ = 0.0547 and β = 0.0596. The number of ports in normal range is 4512, and the number of abnormal ports is 801. Among these ports, there are 176 scanned ports and 625 server failure ports.

We will calculate communication difference cumulative value A of 4512 normal ports, and use two-class EM algorithm to calculate parameter values of client ports and server ports (μ_A, σ_A, μ_B, σ_B, p_A) = (−0.0039, 0.0363, 1.1873, 0.7509, 0.85). Therefore, we can confirm the range of client ports is [−0.1325, 0.1186], the range of server ports is (−2,−0.1325) and (0.1186, 2). Correspondingly, we can calculate the parameters ϕ and φ in Table 2 as ϕ = 0.1325 and φ = 0.1186. Among the normal ports, there are 3875 client ports and 637 server ports.

We classify server ports into three types, overlay ports, server ports and P2P ports. We distinguish the type of client ports by communication difference cumulative value A, and classify overlay ports, i.e. the range of A as (−2,−0.1325) are overlay ports, and the amount is 25. The range of (0.1182, 2) include general server ports and P2P ports, and the amount is 612. In the next, we will use two-class EM algorithm to classify 612 ports with the distribution parameters as (μ_A, σ_A, μ_B, σ_B, p_A) = (0.6966, 0.4845, 1.8111, 0.1091, 0.45). The result is that: the range of server ports is [0.1186, 1.585], the amount is 268; the range of P2P ports is (1.535, 2), and the amount is 344. Thus, we can have the parameter γ = 0.465 (2− γ = 1.535). We put the experiment result of measured parameter δ =0.0547, β = 0.1325, φ = 0.1186, γ = 0.465 into interval formula in Table 2. We can get Figure 10, which shows the interval distribution of four type real normal ports corresponding to Figure 6.

In this test, we need use two-class EM algorithm three times to classify ports. The number of port used in the EM algorithm is 65536, 5313, and 4512 in the measured interval respectively. The maximal number of port is less than 65536. We test the cost of port clustering algorithm in the server equipped with CPU P4 Xeron2.4*2, memory 2 GB, hard disk 1.2 TB, NIC 1000 G, and its running time is 850 ms.

4.3 Experimental analysis

As discussed in Section 1, we know that well-known Ports, which range between 0 to 1023, and are closely bound to some specific services. Registered Ports, port numbers range from 1024 to 49151, they are used by processes of procedures and executed programs of common users. Ephemeral Ports, most implementations use port numbers between 1024 to 5000. Reserved Ports: port numbers larger than 5000 are reserved for other services. We can divided 65536 ports into four types: Well-known Ports, Ephemeral Ports, Registered Ports, Reserved Ports. The role of Well-known port is service port. The role of Ephemeral Port is

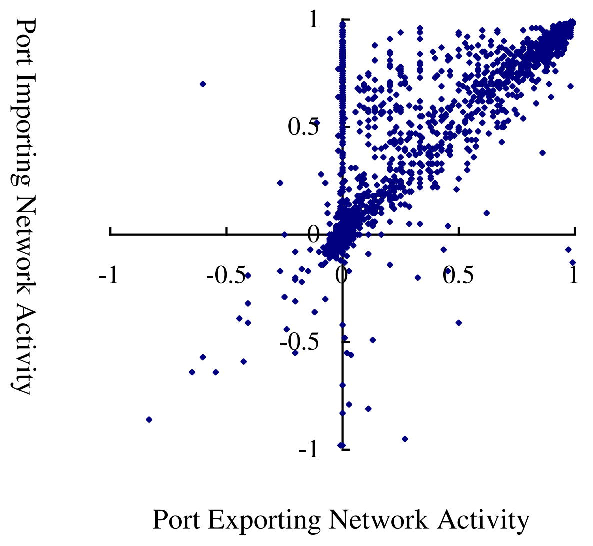

Figure 9 Active ports. It shows the two-dimensional graph of metric of host difference of 5313 active ports.

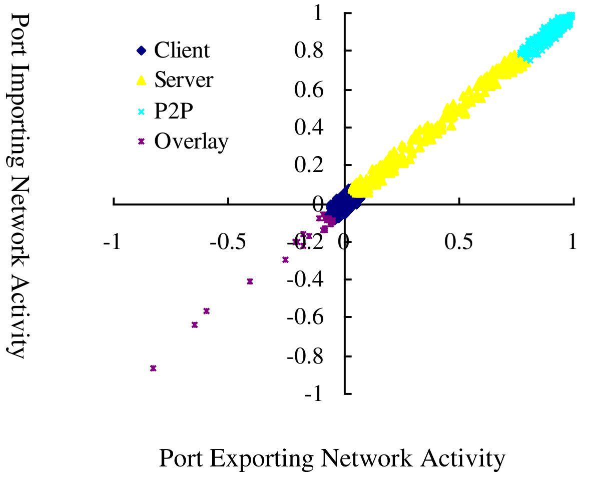

Figure 10 Normal ports. classify normal ports into four types, client ports, overlay ports, server ports and P2P ports.

client port. Registered Ports and Reserved Ports are linked to service port in a local host.

There are 5313 active ports in the experiment. Table 3 gives the relationship between the four port types and the six port roles defined in this paper.

The number of active ports is only 2.14% of the total number of the Well-known ports, which shows that the vast majority of the Well-known port is closed. In Table 3, only seven of the Well-known port is as service port. We can't find any port with client port role in the Well-known port, which also shows that our proposed method can give a correct classification to the client port role. All Ephemeral Ports are active ports, and nearly all Ephemeral Ports are client port role, which is consistent with the definition of Ephemeral Ports.

In the Registered Ports, Service Port and P2P Port account for approximately 50% of the port. Abnormal ports account for another 50%. The result shows that the Registered Ports, port numbers range from 5001 to 49151, is the most common user applications, but also the most vulnerable to attack. Reserved Ports are used nearly.

4.4 Agility validation on PortView

Trace2 was collected only over 5 seconds on May 4, 2012 with 2.82 million packets, 1.89 GB traffic traces, and 17,000 flows. Such a short observation window provides relatively less information to PortView such that we can validate the agility of PortView. We use the activity metric value D to cluster all ports into active and inactive ports. The corresponding five metrics (μ_A, σ_A, μ_B, σ_B, p_A) = (62.34, 325.23, 5.45, 3.08, 0.16). The number of active ports in this experiment is 5215. We calculate communication difference balance value B of all 5215 active Ports. (μ_A, σ_A, μ_B, σ_B, p_A) = (0, 0.527, 0.542, 0.671, 0.55). The number of ports in normal range is 4257, and the number of abnormal ports is 958.

We calculate communication difference cumulative value A of 4257 normal ports. The parameter values of client ports and server ports (μ_A, σ_A, μ_B, σ_B, p_A) = (0, 0.0645, -0.0112, 0.1136, 0.85). Among the normal ports, there are 4151 client ports and 105 server ports. This experiment result shows that PortView can perform effectively even only provided a short observation window.

Table 3 Relationship between the four port types and the six port roles

Port role	Well-known ports	Ephemeral ports	Registered ports	Reserved ports
Total Number of Ports	1024	3977	44151	16384
Client Port	0	3761	29	84
Service Port	7	51	209	2
P2P Port	4	0	340	0
Overlay Port	0	15	9	1
Scanning Port	7	31	135	3
Failed Service Port	4	119	497	5
sum	22	3977	1219	95
Ratio of active ports	2.14%	100%	2.76%	0.58%

5. Related work

Port based traffic analysis can classify application layer protocols [2,3] used port numbers to analyze the type and trend of traffic application. Wei Li et al. [4] made correlation analysis between metrics related to traffic and application types, and found that, correlation between server ports and application types is the strongest of all the traffic metrics, while that between client ports and application types is very weak, so the discovery of server's port has great significance for improving the accuracy of traffic application classification.

McNutt and De Shon [5] analyzed correlation between ephemeral ports and malicious traffic patterns. Wang [6] analyzed the distribution of ports arger than 1024. Jeff Janies [7] studied the time series model of all the 65536 ports in a host, and described the model of server ports and the ports which were attacked by scanning through the visualization of time series graph. DongJin Lee [8] analyzed the distribution of the UDP to TCP ratio that corresponds to the ports. Allman [9] suggested using different methods to choose ephemeral ports to strengthen the security.

Kuai Xu [10] clustered source ports and destination ports, and calculated source IP entropy distribution, destination IP entropy distribution and source or destination port entropy distribution using information entropy, they classified entropy into low, medium, high three types of value, higher entropy indicated number of different corresponding IPs or ports are larger. We can check if a port is a server port or vulnerability by Kuai Xu's Methodology. Thomas Karagiannis [11] believed that since client may use random ports to connect to server ports, if the port number of an IP address equals the number of related flows, the IP host is a client.

The problem in PortView is essentially similar to the challenges in superspreaders detection to maintain all the flows and the IP addresses existent in the memory during the measurement interval. Such as Snort [12] maintains a record for each active connection and a connection counter for each source IP. Because the number of flows and IP addresses are so huge in the network that these algorithms require a lot of memory to maintain the flows and the IP addresses records, or usually detect the superspreaders in low-speed links. In 2005, Venkataraman [13] proposed two algorithms, one-level filtering algorithm and two-level filtering algorithm, which maintain two hash tables to find superspreaders. These filtering algorithms proposed a t-superspreader IP detection algorithm based on sampling from the set of distinct source destination pairs. In this algorithm, t-superspreader IP is defined as more than tN connections to different destination IP addresses, where N is the number of source-destination IP pairs in the measurement interval.

Some data streaming algorithms have been applied to detect superspreaders. In 2007 Noriaki [14] also proposed a parametric algorithm based on the Bloom Filter to measure the flow numbers. However, they don't compensate for the error brought by the hash collisions, and the memory consumption of the Bloom Filter (BF) is very large, so it is difficult to maintain the BF structure in the SRAM to adapt to the high speed links measurement. In 2003, Cohen [15] introduced a Spectral Bloom Filter (SBF), an extension of the original bloom filter to multi-sets, allowing the filtering of elements whose multiplicities are below a threshold. In 2005, Qi [16] proposed two algorithms for the detection of super sources and destinations. The first simple algorithm used a standard hash-based flow sampling algorithm, with a bitmap structure to maintain the flow records. The second algorithm uses a two-dimensional bit array to store the flow record and the flow number. In 2009, MyungKeun [17] created a virtual bit vector for each source by taking bits uniformly at random from a common one-dimension bit array instead of the two-dimensional bit array to store the flow record and the flow number.

Some interesting approach uses graphlet. One of the typical works has been proposed in [11] that can capture host behavior using empirically derived patterns. They represent these patterns using graphs, which they call graphlets. Having a library of these graphlets, they then seek for a match in the behavior of a host under examination. A similar work can also be found in [18], which proposed the use of a graph-based structure which is called a graphlet, to capture the interactions among the transport layer protocols, the destination IP addresses and the port numbers. Guillaume Dewaele [19] compared for two given hosts (with the anonymized IP provided in the trace) trace of the computed 9D features and of the associated graphlets.

The EM algorithm also groups the traffic flows into a small number of clusters and creates classification rules from the clusters in some previous works. Nguyen [20] looked at research into the application of Machine Learning techniques to IP traffic classification. In 2004 McGregor et al. [21] applied the EM algorithm to cluster traffic with similar observable properties into HTTP, FTP, SMTP, IMAP, NTP and DNS traffic. The EM algorithm proposed in AutoClass [22] determines the number of clusters and the parameters that govern the distinct probability distributions of each cluster. The EM algorithm in this paper is different from the previous work because we only classify port traffic into six behavior zones using the EM algorithm rather than some specific traffic applications.

6. Conclusion

Characterizing Macroscopic Port Roles from network traffic provides highly valuable information for various network management tasks. The main purpose of this paper is not to exhaustively list all possible port behaviors, but provides a definition and classification of port roles. In this paper, we firstly classify port roles into six categories, which are scanned ports, failed service ports, client ports, server ports, P2P ports, and overlay ports. Secondly, we propose a port role distribution diagram and define a port behavior measurement metrics to characterize network traffic. A two-class EM fuzzy clustering algorithm is designed to quantitatively determine the classification criteria in the classification distribution diagram. Finally we validate the proposed method based on network traffic captured from a link in the CERNET.

Our contributions include: (1) we research on ports of communication pattern and propose a method to evaluate ports role; (2) we define metrics which can distinguish different ports of communication pattern, and propose a two-class EM fuzzy clustering algorithm, which can achieve fast classifying multi-role of ports; (3) we introduce a new research direction of classifying port role from macroscopic points, which can be used to identify ports type in real-time, discover new applications, and detect new behavior of network attacks.

Competing interest

The authors declare that they have no competing interest.

Authors' contributions

GC created and developed the approach. YT participated in the experiments. GC and YT wrote the manuscript. Both authors read and approved the final manuscript.

Acknowledgment

This work was sponsored by the National Grand Fundamental Research 973 program of China under Grant No. 2009CB320505, the National Nature Science Foundation of China under Grant No. 60973123, the Technology Support Program (Industry) of Jiangsu Province under Grant No. BE2011173, the Qing Lan Project of Jiangsu Province, the Six major talent Summit Project of Jiangsu Province.

Author details

[1]School of Computer Science and Engineering, Key Laboratory of Computer Network and Information Integration, MOE, Southeast University, Nanjing, P.R. China. [2]School of Information Technology, Illinois State University, Normal, IL, USA.

Received: 1 March 2013 Accepted: 3 March 2013
Published: 14 March 2013

References

1. IANA port numbers. http://www.iana.org/assignments/port-numbers
2. Kim H, Claffy K, Fomenkov M, Barman D, Faloutsos M, Lee K (2008) Internet traffic classification demystified: myths, caveats, and the best practices. In: CONEXT '08: Proceedings of the 2008 ACM CoNEXT Conference. ACM, New York, NY, USA, pp 1–12

3. Labovitz C, Iekel-Johnson S, McPherson D, Oberheide J, Jahanian F, Karir M (2009) Internet Observatory Report. http://www.nanog.org/meetings/nanog47/presentations/Monday/Labovitz_ObserveReport_N47_Mon.pdf
4. Li W, Canini M, Moore AW, Bolla R (2009) Efficient application identification and the temporal and spatial stability of classification schema. Computer Networks 53(6):Pages 790–809
5. McNutt J, Shon MD (2005) Correlations between quiescent ports in network flows. Proceedings of FloCon, Pittsburgh, Pennsylvania, USA
6. Wang H, Zhou R, He Y (2008) An Information Acquisition Method Based on NetFlow for Network Situation Awareness. In: Advanced Software Engineering and Its Applications, 2008. ASEA 2008. IEEE. pp 23–26
7. Janies J, Plots E (2008) A Low-Resolution Time Series for Port Behavior Analysis. In: VizSec '08: Proceedings of the 5th international workshop on Visualization for Computer Security. SpringerVerlag, Berlin, Heidelberg, pp 161–168
8. Lee DJ, Carpenter BE, Brownlee N (2010) Observations of udp to tcp ratio and port numbers. In: Internet Monitoring and Protection (ICIMP), 2010 Fifth International Conference on. IEEE. pp 99–104
9. Allman M (2009) Comments on selecting ephemeral ports. SIGCOMM Comput Commun Rev 39(2):13–19
10. Xu K, Zhang ZL (2008) Supratik Bhattacharyya, Internet traffic behavior profiling for network security monitoring. IEEE/ACM Transactions on Networking (TON) 16(6):1241–1252
11. Karagiannis T, Papagiannaki K, Faloutsos M (2005) BLINC: multilevel traffic classification in the dark, ACM SIGCOMM Computer Communication Review. ACM 35(4):229–240
12. Roesch M (1999) Snort-lightweight intrusion detection for networks. In: Proceedings of the 13th USENIX conference on System administration. pp 229–238
13. Venkataraman S, Song D, Gibbons P, Blum A (2005) New streaming algorithms for fast detection of superspreaders. In: Department of Electrical and Computing Engineering. p 6
14. Kamiyama N, Mori T, Kawahara R (2007) Simple and Adaptive Identification of Superspreaders by Flow Sampling, INFOCOM 2007. In: 26th IEEE International Conference on Computer Communications. IEEE. IEEE. pp 2481–2485
15. Cohen S, Matias Y (2003) Spectral Bloom Filters. In: Proceedings of the 2003 ACM SIGMOD International Conference on the Management of Data. ACM Press, New York, pp pp. 241–252
16. Qi Z, Kumar A, Xu J (2005) Joint Data Streaming and Sampling Techniques for Detection of Super Sources and Destinations. In: Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement. USENIX Association. pp 7–7
17. Yoon MK, Li T, Chen S, Peir JK (2009) Fit a Spread Estimator in Small Memory. In: INFOCOM 2009, IEEE. IEEE. pp 504–512
18. Karagiannis T, Papagiannaki K, Taft N, Faloutsos M (2007) Profiling the End Host. In: Passive and Active Network Measurement. pp 186–196
19. Guillaume D, Yosuke H, Pierre B, Kensuke F (2010) Unsupervised host behavior classification from connection patterns. International Journal Of Network Management 00:1–17
20. T. T Nguyen T, Grenville Armitage A (2008) A survey of techniques for internet traffic classification using machine learning. IEEE Communications Surveys & Tutorials 10:56–76, FOURTH QUARTER
21. McGregor A, Hall M, Lorier P, Brunskill J (2004) Flow clustering using machine learning techniques. In: Proc. Passive and Active Measurement Workshop (PAM2004). Antibes Juan-les-Pins, France, pp 205–214
22. Erman J, Arlitt M, Mahanti A (2006) Traffic Classification Using Clustering Algorithms. In: Proceedings of the 2006 SIGCOMM workshop on Mining network data. ACM. pp 281–286

doi:10.1186/1869-0238-4-9
Cite this article as: Cheng and Tang: **PortView: identifying port roles based on port fuzzy macroscopic behavior.** *Journal of Internet Services and Applications* 2013 **4**:9.