MAFIC: Adaptive Packet Dropping for Cutting Malicious Flows to Push Back DDoS Attacks

Yu Chen, Yu-Kwong Kwok, and Kai Hwang
University of Southern California, Los Angeles, CA 90089, USA

***Abstract*— In this paper, we propose a new approach called MAFIC (*MAlicious Flow Identification and Cutoff*) to support adaptive packet dropping to fend off DDoS attacks. MAFIC works by judiciously issuing lightweight probes to flow sources to check if they are legitimate. Through such probing, MAFIC would drop malicious attack packets with high accuracy while minimizes the loss on legitimate traffic flows. Our NS-2 based simulation indicates that MAFIC algorithm drops packets from unresponsive potental attack flows with an accuracy as high as 99% and reduces the loss of legitimate flows to less than 3%. Furthermore, the false positive and negative rates are low–only around 1% for a majority of the cases.**

***Index Terms*—packet dropping policy, DDoS defense, malicious flows, probing, duplicated ACKs.**

I. Introduction

DDoS (Distributed Denial-of-Service) attacks have been the most devastating assaults on the Internet [6], [7], [13], [14]. A DDoS attack is simple to launch—flooding the target host and/or the associated network links with tremendous amount of packets that the victims are incapable to handle. Legitimate traffic is then simply blocked from reaching the victims. A plethora of defense and response mechanisms have been suggested in the literature, including packet traceback [1], packet filtering [9], [11], [16], and pushback [2], [8].

Simply put, tackling DDoS attacks entails handling two critical issues: (1) Accurately identifying the machines participating in forwarding malicious flows; and (2) Incisively cutting off the malicious flows at those machines. For example, in Figure 1, through some response mechanisms (e.g., traceback), the victim may be able to identify the ingress routers (with respect to the domain under protection) that participate in forwarding the malicious flows.

Corresponding author: Kai Hwang, Email: kaihwang@usc.edu, Tel: 213-740-4470, Fax: 213-740-4418. This research was supported by an NSF ITR Research Grant under contract number ACI-0325409. Y.-K. Kwok participated in this project while he was visiting USC, on sabbatical leave from the University of Hong Kong (HKU). Kwok's research at USC was also supported by a research grant from the Research Grants Council of the HKSAR and the 2003-2004 Outstanding Young Researcher Award given by HKU.

Such routers are called Attack Transit Routers (ATRs) [2]. The next step is to instruct these ATRs to preferentially block all malicious flows.

Unfortunately, accurately segregating malicious flows from legitimate flows is very difficult, because of the following two practical issues: (1) IP source addresses of attack packets are inevitably spoofed because both ingress filtering [4] and egress filtering [18] are still far from widely deployed; (2) It is inconceivable that all routers across different autonomous systems (AS's) would cooperate in inserting traceback path information [1].

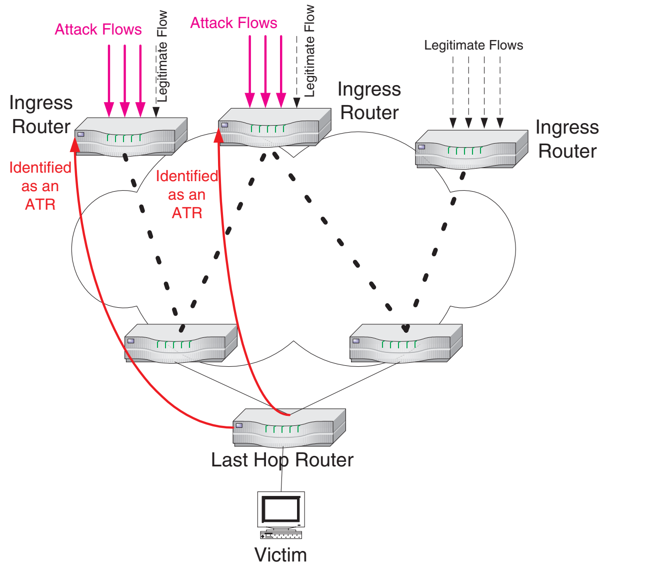


Fig. 1. An integrated DDoS pushback scheme, which identifies the attack-transit routers (ATRs) and drops attack packets adaptively from the malicious flows with spoofed source IP addresses.

In this paper, we propose an adaptive algorithm called *MAlicious Flow Identification and Cutoff* (MAFIC). This algorithm can accurately segregate potential malicious flows from the legitimate flows and then cut these flows off completely. MAFIC works by judiciously issuing lightweight probes to flow sources to check the legitimacy. MAFIC would then drop malicious attack packets with very high accuracy, while minimizes the loss on legitimate traffic flows.

The rest of this paper is organized as follows. Section II presents an overview of our recently proposed set-union counting pushback technique for accurately identifying ATRs. Section III contains the design of our proposed MAFIC algorithm. In Section IV, we describe our NS-2 simulation setup and the performance metrics used. Sections 5 to 8 include the simulation results and their interpretations. Section 9 concludes with some final remarks and suggestions for further work.

II. Review of Our Proposed Set-Union Counting Pushback Technique

In our recent study [2], we have proposed a low-complexity practical technique for identifying the ingress ATRs that unknowingly participate in forwarding malicious flows. Our technique is based on accumulating very light-weight statistics (requiring only $O(\log \log n)$ storage capacity [3]) at each router within the AS such that when an abnormally huge volume of traffic arrives at a particular last hop router (as shown in Figure 1), this victim router can quickly identify the ATRs with very high accuracy.

Specifically, the core network is modeled (i.e., the network of routers) as a directed graph $G = (V, E)$, where V is the set of routers R_i ($i = 1, \ldots, |V|$) and E is the set of directional links l_{ij} ($i, j = 1, \ldots, |V|$). Furthermore, we use the following key notation [10]:

- S_i: The set of packets that originate from a source router R_i (i.e., the packets are injected into the network via this router); and
- D_i: The set of packets that terminate at a destination router R_i (i.e., the packets leave the core network via this router).

Thus, to compute the traffic matrix $A = \{a_{ij}\}$, we need to determine: $a_{ij} = |S_i \cap D_j|$. Observe that the intersection computation can be transformed into a union operation [2]: $a_{ij} = |S_i \cap D_j| = |S_i| + |D_j| - |S_i \cup D_j|$. We can use counters at each router R_i to determine $|S_i|$ and $|D_i|$. However, we also need to compute $|S_i \cup D_j|$ efficiently. This can be done by the stochastic averaging algorithm and the distributed max- merge algorithm [3].

By keeping track of the following values: $|S_i|$, $|D_j|$, and a_{ij}, we can determine whether a last hop router is under an incipient DDoS attack. Specifically, if we find that the value of $|D_j|$ is abnormally high, then we can infer that the router R_j is under a raging DDoS attack. Then, we can identify the ATRs by checking the values of a_{ij} for all ingress routers i.

However, one problem remains unsolved in our previous study is the next step–how to incisively blocking all the malicious flows at such ATRs. Indeed, in [2] we only used a simple proportionate packet dropping approach, i.e., all packets, legitimate or malicious, are dropped with the same probability. This is obviously not a satisfactory approach because of the inevitable "collateral damages" incurred.

III. MAFIC: Adaptive Malicious Flow Identification and Cutoff

In this section, we first describe the design rationale of our proposed MAFIC algorithm, followed by a detailed description of its packet dropping mechanisms.

A. Design Rationale

Our MAFIC algorithm handles two of major challenges in identifying DDoS attack flows: source IP spoofing and multiple distributed zombies utilization. There are two scenarios at the extreme of the spectrum of IP spoofing. On one hand, all source IP addresses are illegal or unreachable. On the other hand, all the attack packets carry legitimate source IP addresses. Note that "legitimate" means that the source IP addresses of a packet is a valid address of a certain subnet within a certain AS. It does not mean that the source IP address represents the true IP address of the machine that launches the attack flows.

Our algorithm addresses DDoS attacks lying somewhere in between these two extreme cases, that is, some IP addresses are bogus but some other are "legitimate" in the above sense. For packets with illegal or unreachable source IP addresses, we place them in a data structure called *Permanently Drop Table* (PDT) and drop all such kind of packets. The rationale is that although probably they are not part of the DDoS attack we are fighting against, we believe that they do not belong to any normal application. Furthermore, for packets carrying legitimate source IP addresses, we use it as part of the label of a flow instead of taking count on that to figure out the sender. Then, all packets with the same label will be treated as members of the same flow.

Many traffic management algorithms identify misbehaving flows by monitoring and comparing their bandwidths with other flows sharing the link [5], [17]. One problem arises when apply such an idea in packets classification against DDoS attacks. As multiple zombies are used to launch the attack concurrently, the traffic from each single zombie may behave well by occupying bandwidth shares similar to general TCP like traffics, or even less than that. Therefore, MAFIC algorithm adopts an adaptive packets dropping policy that segregates the malicious flows from others, and minimizes the impact on normal users flows. On receiving the notification of DDoS attack from the victim router, each ATR begins dropping packets with certain probability and send duplicated ACKs to hosts with source IP address. A *Suspicious Flow Table* (SFT) is established and flows of those dropped packets are recorded and a timer starts.

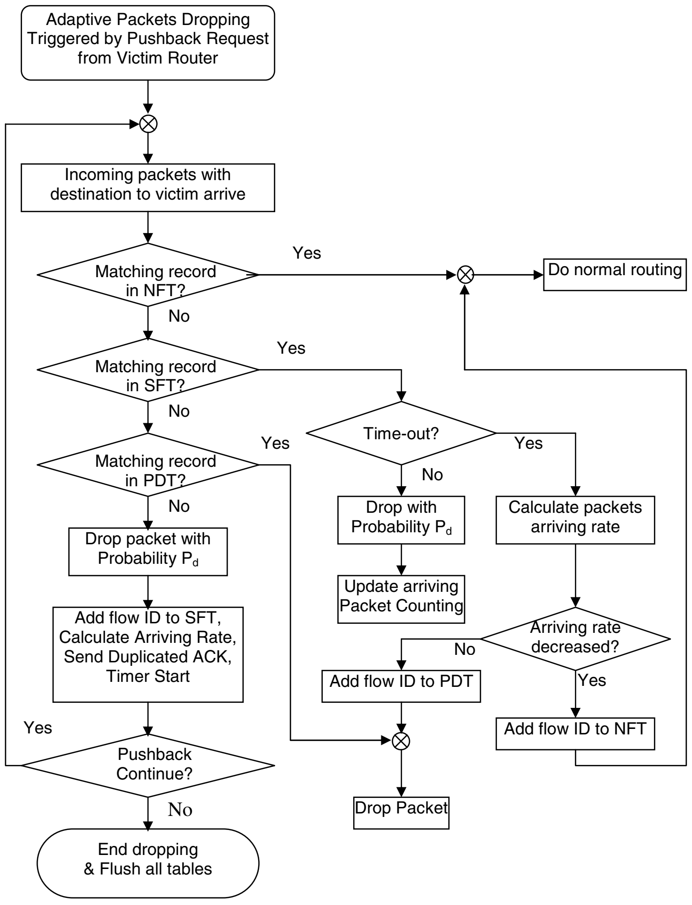
Meanwhile, the identified ATRs keep watching variances of arrival rate of these suspicious flows. If the arrival rate of a flow decreases accordingly, it would be moved from the SFT to the *Nice Flow Table* (NFT) as it behaves in a TCP-friendly manner; otherwise, the unresponsive flows will be added into PDT. Although not all unresponsive flows originate from malicious attacks, their non-TCP-friendly behaviors would not be helpful to relief the victim from the flooding of packets. Since TCP based flows consist more than 80% of network traffics [12] and major parts of attacks use TCP protocol [14], the cost of collateral damage on unresponsive legitimate traffic flows is acceptable.

B. The MAFIC Algorithm

In this section, we describe the detailed mechanisms of our proposed MAFIC algorithm. Although source IP addresses are generally spoofed in attacking packets, they give us some useful information. Thus, the 4-tuple {`Source IP, Destination IP, Source Port, Destination Port`} is used as a label to used to mark each flow in tables. Thus, in case multiple flows originate from the same sender, each flow has a unique flow ID and its own entry in SFT, NFT and PDT.

To minimize the storage overhead incurred by the extra lists added to implement our adaptive packets dropping algorithm, we store only the output of a hash function with the label as the input instead of the label itself. Figure 2 shows the control actions of the MAFIC algorithm. One critical issue in the algorithm is the value of the timer used to check if the flow behaves according to the packet drop. Fortunately, RTT information is available in most TCP traffic flows by checking the time stamp in the packet header.

To allow for a moderate amount of time for the legitimate sources to respond, we set the timer equal $2 \times$ RTT. While a packet is dropped, the current time stamp is added into the SFT along with the hashed label. Once there are more packets belonging to this flow arrive, the time stamps are compared with the record and the arrival rate is updated. If a flow is found having lower sending rate while the timer expires, it would be moved to NFT and no more dropping will be done against its packets. On the other hand, if the sending rate is not decreased, the flow would be moved to PDT and all its future packets are dropped.



Adaptive Packets Dropping Triggered by Pushback Request from Victim Router
Incoming packets with destination to victim arrive
Matching record in NFT?
Yes
Do normal routing
No
Matching record in SFT?
Yes
No
Time-out?
Yes
Matching record in PDT?
Yes
No
No
Drop with Probability P_d
Calculate packets arriving rate
Drop packet with Probability P_d
Update arriving Packet Counting
Add flow ID to SFT, Calculate Arriving Rate, Send Duplicated ACK, Timer Start
Arriving rate decreased?
No
Add flow ID to PDT
Yes
Yes
Add flow ID to NFT
Pushback Continue?
No
Drop Packet
End dropping & Flush all tables

Fig. 2. The control flow of our proposed MAFIC algorithm. (NFT: Nice Flow Table, SFT: Suspicious Flow Table, PDT: Permanently Drop Table).

IV. Simulation Setup and Performance Metrics

We have implemented MAFIC algorithm in the NS-2 simulator [15]. A subclass of Connector named as `LogLogCounter` is added to the head of each `SimplexLink` and a `TrafficMonitor` is coded into the simulator to compute the traffic matrices [2]. The `LogLogCounter` class is used to compute the summary of distinct packets traversed through each router. When a packet is forwarded to a particular link by the routing module on each node, the `recv` method of `LogLogCouter` object associated with this link will be called. While triggered by the victim node, the `LogLogCounter` object begins to tentatively drop incoming packets targeting to the victim with certain probability, and put the flow into the SFT.

Then it keeps monitoring the packets arrival rate for a time period of $2 \times$ RTT. If the incoming rate of the flow decreases accordingly, the `LogLogCounter` object moves the flow into the NFT and will not drop any packets belonging to this flow in the future; otherwise, it will put the flow into the PDT and all packets of that flow will be considered as malicious attacks and dropped. The `TrafficMonitor` keeps track of all `LogLogCounter` objects and for each time period, it will be triggered to compute the traffic matrix for this time period using the set-union counting algorithm. Our notation used throughout our simulation results is listed in Table I. The default settings of simulation parameters are listed in Table II.

TABLE I
DEFINITION OF NOTATION

Symbol	Definition
P_d	SFT packet dropping probability
R	Flow rate (packets/second)
V_t	Traffic volume (total number of flows)
Γ	Percentage of TCP flows
α	Attacking packets dropping accuracy
N	Domain size (number of routers)
β	Traffic reduction rate
θ_p	False positive rate
θ_n	False negative rate
L_r	Legitimate packets dropped rate in identifying malicious flows

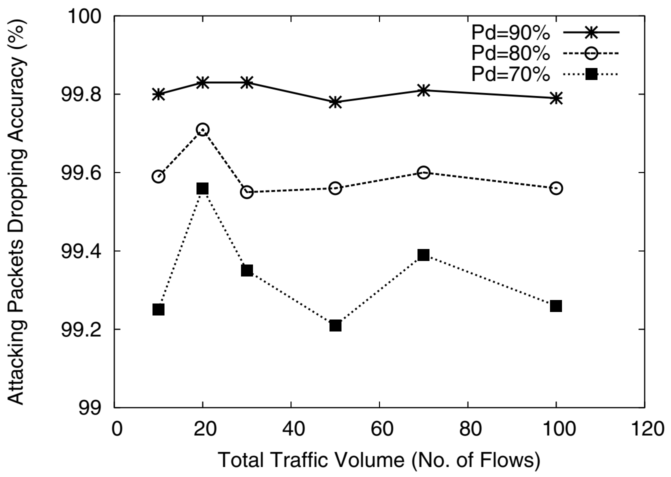
TABLE II
DEFAULT SETTING OF PARAMETERS

Parameter	Default Value
P_d	90%
R	1×10^6 packets/second
V_t	50 flows
Γ	95%
N	40 routers

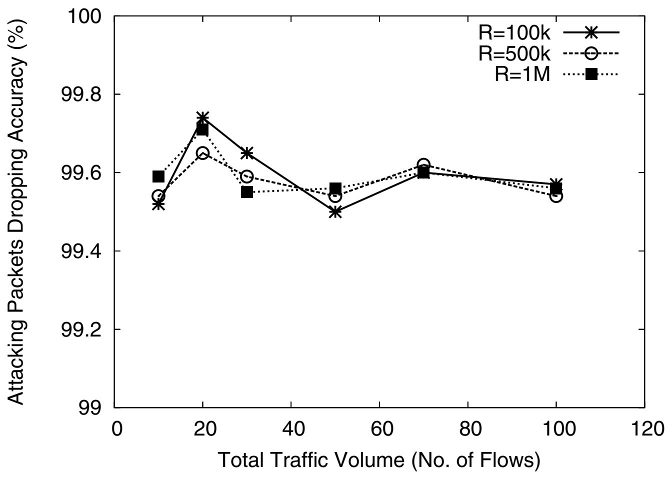
V. PERFORMANCE RESULTS

A. Attacking Packet Dropping Accuracy

In this section, we present our results on attacking packet dropping accuracy. This parameter indicates how accurate the SFT/PDT mechanism can identify the attack packets and drop them. Specifically, we define the accuracy as the percentage of dropped malicious packets among total number of malicious packets that arrive at the ATRs. Figure 3 shows the attacking packet dropping accuracy under different traffic volumes. We can see that the MAFIC algorithm is robust because the accuracy is consistently within the range of 99.2% to 99.8% even under dramatically different traffic conditions.



(a) Under three packet dropping probabilities: 70%, 80%, and 90%



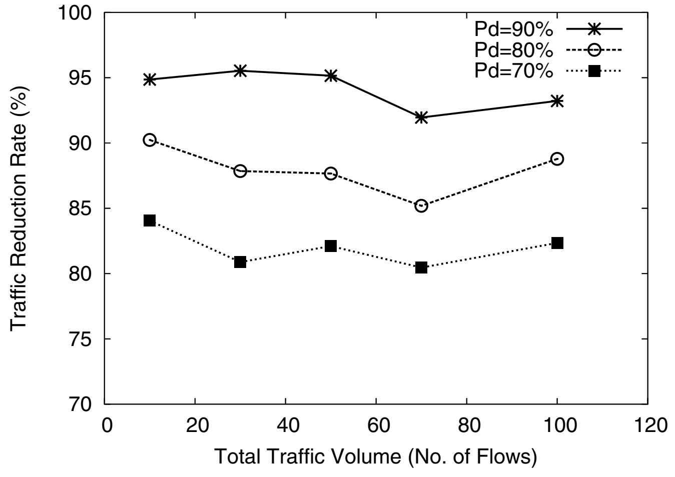
(b) Under three source packet sending rates from 100kbps to 1Mbps

Fig. 3. Attack packet dropping accuracy of the proposed MAFIC algorithm under different traffic conditions.

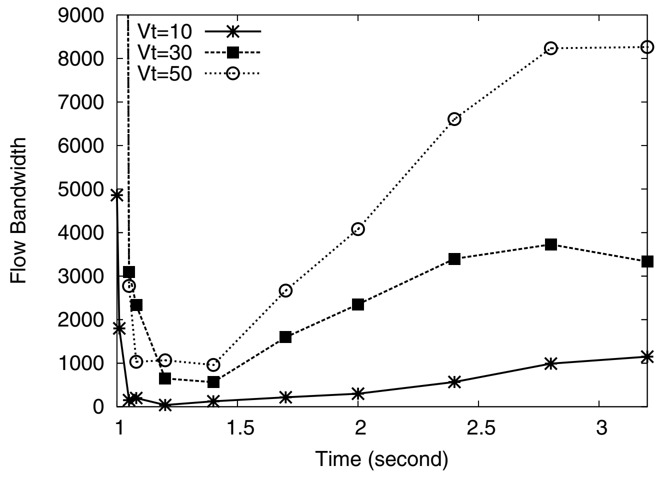
B. Performance of Flow Cutting

To fighting against DDoS attacks, the responsiveness of the defense scheme is one of the most critical considerations. Figure 4 shows the results of variations of flow bandwidth before and after the MAFIC algorithm is invoked. We can see that as soon as the MAFIC algorithm is triggered, routers begin to drop packets having destination addresses matching the victims address.

As shown in Figure 4(a), arrival rate at victim is cut off very quickly within a time period of $2 \times$ RTT. By setting the dropping probability to 90%, the flows targeting at the victim link are reduced to about 95%. The results also indicate that the arrival rates are cut down by 85% and 80% while we set the dropping probability equals to 80% and 70%, respectively. In Figure 4(b), 95% of flows in total traffic volume are TCP flows. We can see that packet rates decrease drastically as soon as our adaptive dropping algorithm triggered into its probing phase. It is important to note that after the cutting of potential attacking flows (i.e., unresponsive flows), the legitimate traffic flows regain their bandwidth shares after passing the test in the probing phase. These results indicate that MAFIC algorithm responses quickly and possesses good scalability.



(a) Under three packet dropping probabilities: 70%, 80%, and 90%



(b) Under three different number of flows: 10, 30, and 50

Fig. 4. Traffic reduction rates and flow bandwidth variations of the proposed MAFIC algorithm under different traffic conditions.

C. Effects on False Alarm Rates

False positive rate (θ_p) and false negative rate (θ_n) are important QoS parameters to characterize the performance of detection algorithms against network attacks. False positive rate is defined as the percentage of legitimate packets wrongly dropped as malicious attacking packets out of the total traffic packets. False negative rate is the percentage of attacking packets that were not dropped or hidden in the traffic to hit the victim node across the defense line without being detected. While θ_p indicates the sensitivity and accuracy of the detector, it also evaluates the impact on well-behaved applications. On the other hand, θ_n reflects the effectiveness of the algorithm against malicious attacks. There is always a tradeoff between the intrusion detection rate and the false alarm rate.

Our simulation results on false alarm rates are plotted in Figures 5 and 6. We can see that the values of θ_p are very low–bounded above by 0.06% for all three cases under different traffic conditions. In general, θ_p increases with the total traffic volume and percentage of TCP flows. But θ_p ceases to increase and even starts to drop when the dropping probability (P_d) becomes very high. Similarly, θ_n drops for high TCP or high total traffic volumes. In general, θ_n reduces with increasing traffic volume. But θ_n increases again when the traffic exceeds a certain limit.

D. Dropping Rate of Legitimate Packets

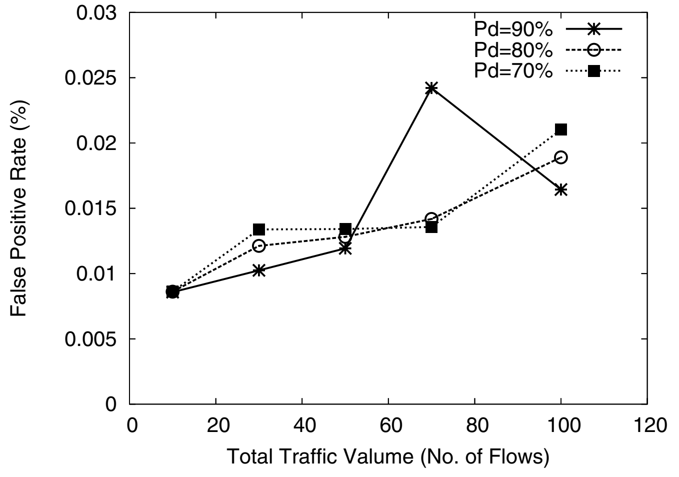
The legitimate-packet dropping rate (L_r) is defined as the number of packets in well-behaved flows dropped during the probing phase and the subsequent dropping process. L_r indicates the degree of severity of the inevitable "collateral damage". Figure 7 shows the results of L_r during the probing phase corresponding to different packet dropping probabilities. We see that even with very high dropping probability (70%, 80%, and 90%), the costs of legitimate flows are insignificant. Furthermore, as the total traffic volume grows, the L_r tends to be stable and converges to the value of about 1%.

VI. Conclusions

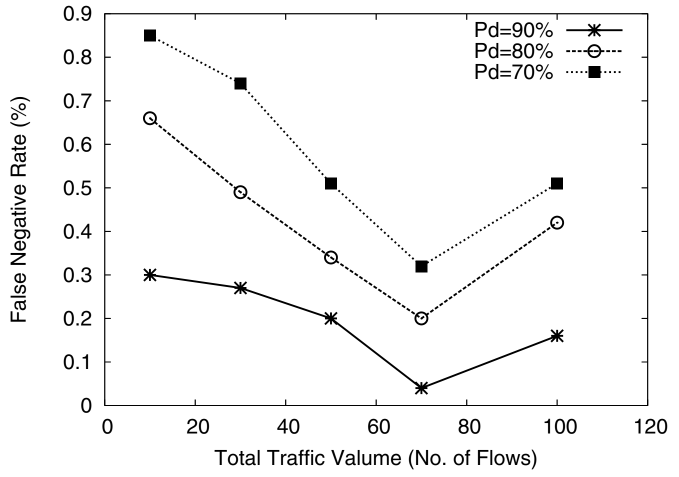
In this paper, we have proposed a novel scheme for malicious flow identification and cutoff called MAFIC. We implemented the adaptive packet dropping and probing algorithm. By monitoring the response to packet loss from the flow source, malicious attacking flows are accurately identified and all their packets are then dropped before reaching the victim. Our extensive NS-2 based simulation results shows that MAFIC responses quickly to reduce the traffic rate by limiting flow bandwidth incisively. Indeed, MAFIC can identify and cut off malicious flows with very high accuracy and minimized impact on the performance of legitimate application flows, especially when major applications use TCP protocol.

Acknowledgments

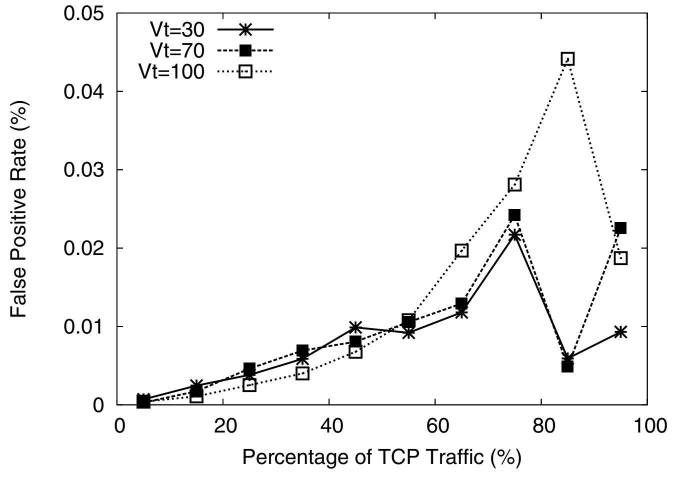
We would like to thank Mr. Min Cai of the Department of Computer Science, University of Southern California, who lets us to use his NS-2 DDoS defense simulator. His generous assistance is also appreciated in conducting the simulation experiments.



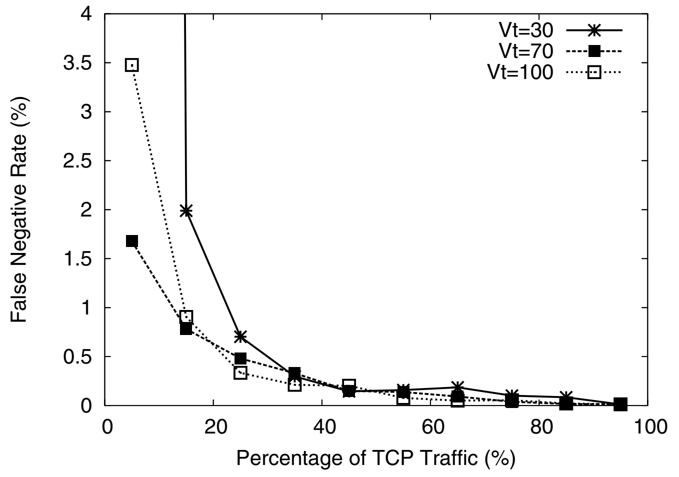
(a) θ_p under three packet dropping probabilities: 70%, 80%, and 90%



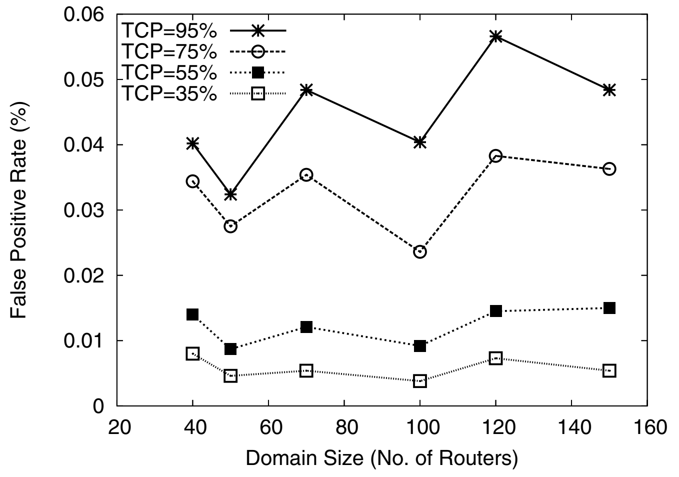
(a) θ_n under three packet dropping probabilities: 70%, 80%, and 90%



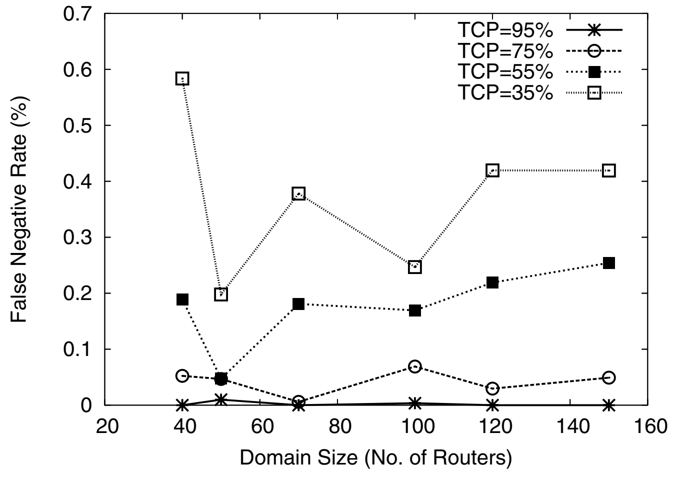
(b) θ_p under three different number of flows: 30, 70, and 100



(b) θ_n under three different number of flows: 30, 70, and 100



(c) θ_p under four different percentage of TCP flows: from 35% to 95%



(c) θ_n under four different percentage of TCP flows: from 35% to 95%

Fig. 5. False positive rates of MAFIC algorithm under different traffic conditions.

Fig. 6. False negative rates of MAFIC algorithm under different traffic conditions.

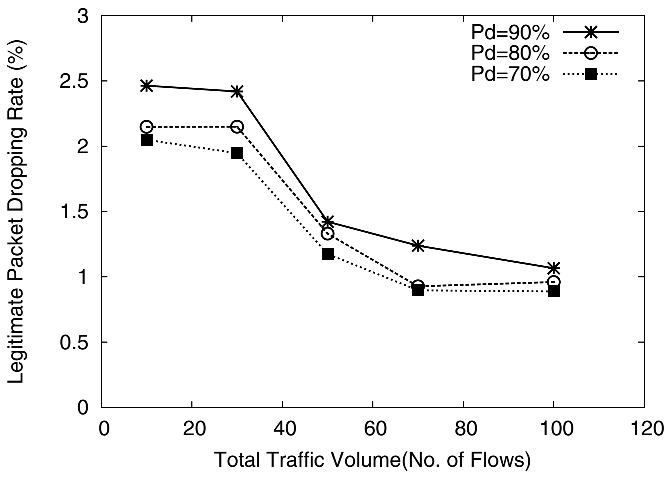


Fig. 7. L_r rates for three packet dropping probabilities.

REFERENCES

[1] H. Aljifri, "IP Traceback: A New Denial-of- Service Deterrent," *IEEE Security and Privacy Magazine*, May/June 2003, pp. 24-31.

[2] M. Cai, Y. Chen, Y.-K. Kwok, and K. Hwang, "A Scalable Set-Union Counting Approach to Pushing Back DDoS Attacks," USC GridSec Technical Report TR-2004-21, Oct. 2004.

[3] M. Durand and P. Flajolet, "LogLog Counting of Large Cardinalities," *Proc. European Symposium on Algorithms 2003*.

[4] P. Ferguson and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing," IETF RFC 2827, May 2000.

[5] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-Based Congestion Control for Unicast Applications," *Proc. ACM SIGCOMM 2000*.

[6] A. Hussain, J. Heidemann, and C. Papadopoulos, "A Framework for Classifying Denial of Service Attacks," *Proc. ACM SIGCOMM 2003*.

[7] K. Hwang, Y. Chen, and H. Liu, "Defending Distributed Systems against Malicious Intrusions and Network Anomalies," *Proc. of the 1st International Workshop on Systems and Network Security (SNS 2005)*, in conjunction with IPDPS 2005, Apr. 2005.

[8] J. Ioannidis and S. M. Bellovin, "Implementing Pushback: Router-Based Defense Against DDoS Attacks," *Proc. NDSS 2002*.

[9] Y. Kim, W. C. Lau, M. C. Chuah, and H. J. Chao, "PacketScore: Statistics-Based Overload Control Against Distributed Denial-of-Service Attacks," *Proc. INFOCOM 2004*.

[10] M. Kodialam, T. V. Lakshman, and W. C. Lau, "High-speed Traffic Measurement and Analysis Methodologies and Protocols," Bell Labs Technical Memo, Aug. 2004.

[11] Y. K. Kwok, R. Tripathi, Y. Chen, and K. Hwang, "HAWK: Halting Anomalies with Weighted Choking to Rescue Well-Behaved TCP Sessions from Shrew DoS Attacks," submitted to the *International Conference on Computer Networks and Mobile Computing (ICCNMC 2005)*, Feb. 2005.

[12] K.-C. Lan, A. Hussain, and D. Dutta, "The Effect of Malicious Traffic on the Network," *Proc. PAM 2003*, April 2003.

[13] J. Mirkovic and P. Reiher, "A Taxonomy of DDoS Attack and DDoS Defence Mechanisms," *ACM Computer Communications Review*, vol. 34, no. 2, Apr. 2004.

[14] D. Moore, G. M. Voelker, and S. Savage, "Inferring Internet Denial-of-Service Activity," *Proc. USENIX Security 2001*.

[15] Network Simulator 2 (NS-2), http://www.isi.edu/nsnam/ns/, 2005.

[16] K. Park and H. Lee, "On the Effectiveness of Route-Based Packet Filtering for Distributed DoS Attack Prevention in Power-Law Internets," *Proc. ACM SIGCOMM 2001*.

[17] P. Pradhan, T. Chiueh, and A. Neogi, "Aggregate TCP Congestion Control Using Multiple Network Probing," *Proc. ICDCS 2000*.

[18] SANS Web Pages, "Egress Filtering," http://www.sans.org/y2k/egress.htm, Feb. 2000.

AUTHORS' BIOGRAPHICAL SKETCHES

Yu Chen received his B.S and M.S in Electrical Engineering from Chongqing University, China, in 1994 and 1997, respectively. He is presently pursuing the Ph.D. degree in Electrical Engineering at the University of Southern California. His research interest includes Internet security, DDoS attacks, automated intrusion detection and response systems, and distributed security infrastructure for Grid Computing environment. He can be reached at *cheny@usc.edu*.

Yu-Kwong Kwok is an Associate Professor in the Department of Electrical and Electronic Engineering, University of Hong Kong (HKU). Dr. Kwok is currently on sabbatical leave from HKU and is a Visiting Associate Professor at USC. His research interests include secure Grid computing, mobile computing, wireless communications, network protocols, and distributed computing algorithms. He received the 2003-2004 Outstanding Young Researcher Award from HKU. Dr. Kwok is a Senior Member of the IEEE. He is also a member of the IEEE Computer Society, IEEE Communications Society, and the ACM. He can be reached at *ykwok@hku.hk*.

Kai Hwang is a Professor and Director of Internet and Grid Computing Laboratory at the University of Southern California. He received the Ph.D. from the University of California, Berkeley. An IEEE Fellow, he specializes in computer architecture, parallel processing, Internet and wireless security, and distributed computing systems. He has authored or coauthored 7 scientific books and 180 journal/conference papers in these areas. Hwang is the founding Editor-in-Chief of the *Journal of Parallel and Distributed Computing*. Currently, he is also an Associate Editor of the IEEE Transactions on Parallel and Distributed Systems. He has performed advisory and consulting work for IBM Fishkill, Intel SSD, MIT Lincoln Lab., ETL in Japan, and GMD in Germany. Presently, he leads the NSF-supported ITR GridSec project at USC. The GridSec group develops security-binding techniques for trusted job scheduling in Grid computing, distributed IDS and push-back of DDoS attacks, fuzzy-logic trust models and selfish Grid Computing models, and self-defense software systems for protecting network-centric computing resources. Professor Hwang can be reached at *kaihwang@usc.edu* or through the URL: http://GridSec.usc.edu/Hwang.html.