International Journal of Software Engineering & Computer Systems (IJSECS)
ISSN: 2289-8522 ,Volume 1, pp.93-105, February 2015
©Universiti Malaysia Pahang
DOI: http://dx.doi.org/10.15282/ijsecs.1.2015.8.0008

INVESTIGATION MODEL FOR DDOS ATTACK DETECTION IN REAL-TIME

Abdulghani Ali Ahmed

Faculty of Computer Systems & Software Engineering, Universiti Malaysia Pahang,
26300 Gambang, Pahang, Malaysia
Email: abdulghani@ump.edu.my

ABSTRACT

Investigating traffic of distributed denial of services (DDoS) attack requires extra overhead which mostly results in network performance degradation. This study proposes an investigation model for detecting DDoS attack in real-time without causing negative degradation against network performance. The model investigates network traffic in a scalable way to detect user violations on quality of service regulations. Traffic investigation is triggered only when the network is congested; at that exact moment, burst gateways actually generate a congestion notification to misbehaving users. The misbehaving users are thus further investigated by measuring their consumption ratios of bandwidth. By exceeding the service level agreement bandwidth ratio, user traffic is filtered as DDoS traffic. Simulation results demonstrate that the proposed model efficiently monitors intrusive traffic and precisely detects DDoS attack.

Keywords: QoS regulations; RED-enabled gateways; SLA violations; DDoS

INTRODUCTION

Every day, Internet attackers come up with new sophisticated tactics to increase their ability of disrupting users' online services. One of their tactics is generating unsolicited traffic in immense volume for overwhelming network resources. A sudden surge in network traffic mostly occurs due to huge traffic generated from single or distributed sources to prevent legitimate users from using network resources, e-businesses, or online services. Distributed denial of services (DDoS) is a sophisticated attack uses huge traffic to overwhelm network resource and deny services to network users(Gyanchandani2, 2010; Jose, 2008; Jaeyeon et al., 2002).

Numerous studies have attempted to devise an effective solution to the DDoS traffic without inflicting harm on normal traffic (Choffnes, 2010; Xuan, 2010). However, monitoring DDoS traffic still has the challenges of capability, scalability, and reliability, which severely hinder the researchers from developing a successful defense. Capability challenge arises due to the high similarity between DDoS and legitimate traffic. Scalability challenge arises because the continuous monitoring of a large-scale network in a scalable manner requires extra overhead in a volume, which mostly results in performance degradation. Reliability challenges arise when the algorithm fails to investigate user traffic in whole or in part. Such a case can be caused by a single point of failure in making filtration decisions.

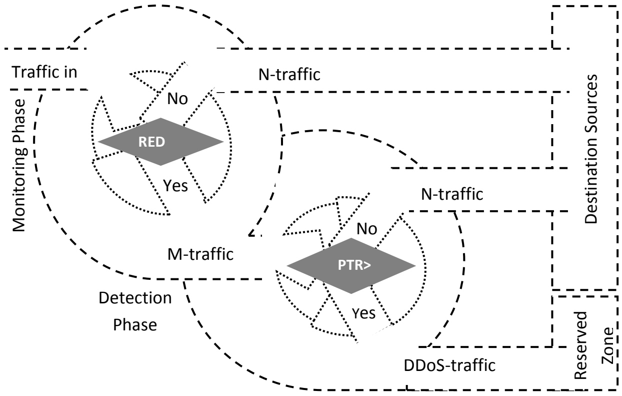

Figure 1. Phases of traffic investigation.

In the detection phase, the *M-traffic* is thus subjected to eventually undergo further investigation. Using passive measurement, the bandwidths consumed by *M-user* are measured as packet transmission rate (PTR) aggregation. The *M-users* that exceed SLA PTR ratios are considered as *attacker*, and their traffic as *DDoS traffic*.

The traffic of users that do not violate the SLA PTR ratios is considered as normal traffic (*N-traffic*).The rest of this study is organized as follows: Section 2 discusses the related works. Section 3 describes the architecture of the filtration model. Section 4 describes the DDoS traffic filtration. Section 5 presents the experimental results and analytical evaluation. Finally, Section 6 concludes and discusses the possibilities for future work.

RELATED WORKS

In general, there are two main categories of IDS, each with its own disadvantages: misuse-based, and anomaly based detection systems. In the main drawback of the first category is its inability to detect new intrusions still unknown to the intrusion detector. Thus, the security policy of these approaches should add new rules when a new type of attack is discovered. The disadvantage of the second category is the possibility of deviation the normal traffic from its distribution pattern signatures. In addition to the existing IDS of misuse-based and anomaly-based, quality of service (QoS)-based IDS is an important subcategory to detect DDoS attacks.

Numerous real-time studies have been conducted regarding the impact of user behavior on QoS regulations to detect DDoS traffic. This study focuses on the related works that use SLA ratios as thresholds to differentiate the DDoS traffic from the normal ones. Important and recent schemes on the SLA monitoring can be found in (Habib et al., 2005; Ahsan et al., 2004; Abdulghani et al,2010; Abdulghani et al, 2011; An, 2006). In these schemes, traffic investigation requires separate communication between the domain edges and SLA management for each delay or jitter, loss, and PTR metrics. As a result, processing and communication overhead may increase, representing a shortcoming for these schemes.

Moreover, to detect SLA violation, delay is estimated in the arrival of packets. Estimation is conducted either by examining the timestamps of probe packets or by dividing the round-trip time (RTT) by two. In the former, the two ends cannot be synchronized. In the latter, the major weakness lies in approximated ratios as a result of asymmetric links in domains. Another study has revealed the difficulty of using the core-assisted scheme because of the high overhead incurred in monitoring core routers (Abdulghani et al, 2011). Furthermore, bandwidth measurements are executed at egress routers rather than at ingress routers. In this case, simultaneous global attacks may not be detected because local attacks get camouflaged.

Additionally, these schemes use central management units for gathering the measured values of users from various domain edges and for making final decisions on traffic investigation. These algorithms may be vulnerable to single points of failure. By contrast, the algorithm in (Ahsan et al., 2004) uses distributed management units on various edges for gathering the measured values of users and filtering DDoS traffic. Although the distributed algorithm is immune against single points of failure, this algorithm is not sufficiently scalable because of the high processing and communication overhead generated in gathering distributed measured ratios.

In the proposed model, an SLA violation is deduced when RED-enabled gateways begin to generate congestion notifications toward misbehaving users. Therefore, investigating the jitter and packet loss ratios is unnecessary. Thus, the proposed model enhances communication overhead by decreasing the exchange of messages and processing overhead by reducing the fraction of traffic, which should be examined.

ARCHITECTURE OF THE INVESTIGATION MODEL

The next subsections describe in detail the agent architecture of the present model.

VIOLATION-MONITOR AGENT

The Violation-monitor agent is used to recognize *M-users* notified by RED technique and to probe the *M-traffic* for PTR measurement. Customers of QoS networks have SLA guarantees from their service providers for delay, jitter, loss, and bandwidth metrics (Tham and Y. Liu, 2005). These QoS metrics are measured to determine abnormal activities in the network. In this paper, delay and packet loss estimation are deduced by monitoring RED-gateway queues.

Traffic policing is executed at domain gateways. The RED technique prevents traffic bursts at gateways by monitoring traffic shifts in the AQSs. Once the AQS of a particular gateway exceeds a predefined threshold, RED notifies the users connected to that edge to reduce the volume of sent packets. The RED algorithm computes the AQS for every packet received at the gateway queue by using a low-pass filter algorithm that uses the exponential weighted moving average (EWMA) technique described in (Gu, 2003). According to (Floyd and Jacobson, 1993), the RED policy uses the EWMA technique to smooth possible short-term increases in queue size (QS) that result from normal traffic bursts or from transitory congestion, thus resulting in a significant increase in the AQS. Therefore, the AQS of the burst gateway is calculated as follows:

$$AQS = AQS \times (1 - w_q) + w_q \times q \tag{1}$$

where q is the instantaneous buffer size of the gateway queue, and w_q is an exponential weight coefficient that defines the time of the low-pass filter with a ratio much less than one. Choosing an appropriate ratio for w_q is important for efficiently calculating the AQS. Thus, if the w_q ratio is too large, the averaging transaction may not filter the transitory congestion at the gateway queue. However, if the w_q ratio is set too low, the response of the AQS to shifts in the actual QS will be too slow; thus, the gateway may not detect the initial levels of congestion. As described in (Floyd and

Jacobson, 1993), the ratio of coefficient w_q is chosen based on the number of packet arrivals at the gateway queue. Therefore, the ratio of w_q should be set to satisfy the following equation:

$$L+1+\frac{(1-w_q)^{L+1}-1}{w_q}<\min_{th} \tag{2}$$

where L is number of packets arriving at the gateway queue, and min$_{th}$ is the RED minimum threshold. In one RTT, the minimum and maximum thresholds, min$_{th}$ and max$_{th,}$, are defined by considering that max$_{th}$–min$_{th}$ must not be less than the typical augment in the AQS. The AQS is then compared with the max$_{th}$ and min$_{th}$ thresholds. If the calculated AQS is less than min$_{th}$, all packets are allowed to pass to the destination. However, if the value of the AQS is greater than the max$_{th}$, the packets are marked to drop. In between, every received packet is marked with an RED-notification in commensurate probability.

VIOLATION-VERIFIER AGENT

The Violation-verifier agent measures the PTR fractions of *M-users* to verify if the consumed ratio exceeds the ratio guaranteed in the SLA. PTR of *M-user* at each ingress edge can be adequately measured by counting the average number of packets generated by a user. According to (Ningning, 2003), the PTR can be accurately measured by multiplying the total generated packets with packet size. Thus, the bandwidth consumed by every *M-user* is computed by measuring the PTR of that user at each ingress edge:

$$PTR^{i}_{M-user}=\frac{packet\_size\times 8\times\left(avg\_sent^{i}_{M-user}\right)}{\Delta t} \tag{3}$$

where $avg\_sent^{i}_{M-user}$ is the average packets sent by the *M-user* at ingress edge *i*, and Δt is the time interval. Conclusions on DDoS traffic sources are made once the end-user sends more than its preset bandwidth share in the SLA.

SLA-MANAGER AGENT

The SLA-manager agent provides a mechanism for managing the detection of DDoS attack.This agent computes the average ratios of the PTR for *M-user* on the basis of user details gathered from various ingress edges; therefore, DDoS traffic is differentiated from legitimate traffic by recognizing misbehaving users who exceed the SLA bit rate of the PTR. The SLA-manager agent is a central unit taking decisions based on congestion reports received from distributed RED gateways. It is activated once it is probed by the Violation-monitor agent. The SLA-manager module is shown in Fig 2 as a separate entity installed in a separated router. In this paper, once the *M-traffic* notified with RED notifications, every ingress edge measures the PTRs of the *M-users* and reports them to the SLA-manager. The SLA-manager add their reported PTR_{M-user} to the aggregated PTR$_{\text{M-user}}$ by using the following formula:

$$PTR_{M-user}=PTR_{M-user}+PTR_{M-user} \tag{4}$$

The process of aggregating the PTR and sending the subtotal to their right and left neighbors is repeated for SLA-manager. Figure 2 shows how SLA-manager exchanges PTR messages with each network edges to resolve the decision of PTR violations. For every *M-user*, the total number of PTR is computed by aggregating the PTR fractions from the SLA-manager. To compare the PTR measured ratios with the SLA bandwidth shares and resolve the decision of PTR violations, the SLA-manager transfers the total $\mathrm{PTR}_{\text{M-user}}$ to percentile ratios by using the following formula:

$$Percentile\_PTR_{M-user} = \left(\frac{PTR_{M-user}}{Bandwidth_{Link}}\right) \times 100\% \qquad (5)$$

where $Bandwidth_{Link}$ is the domain links bandwidth. Therefore, the SLA-manager compares the PTR percentile of *M-user* with the SLA bandwidth share to detect possible DDoS traffic.

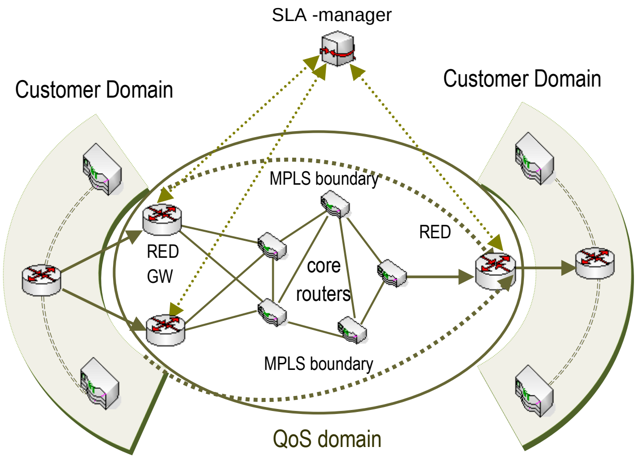

Figure 2. Architecture of the investigation model.

DDOS TRAFFIC INVESTIGATION

Typically, at the ingress routers, user traffic is classified and policed to ensure that each user does not exceed the delay, jitter, packet loss, and bandwidth ratios predefined in the SLA. A user cannot send a volume of packets higher than the SLA rates through a single ingress; however, the user can send a higher volume by sending a volume lower than the SLA bit rate through multiple ingress edges (Habib et al., 2003). Thus, inspection of QoS metrics at each edge does not detect network attacks; however, by doing so at all edges, network attacks can be detected.

Moreover, optimally deploying the model agents is necessary to guarantee the production of an early warning and the deterrence of network attacks. As illustrated in Fig. 2, Violation-monitor and Violation-verifier agents are deployed at ingress routers. However, the SLA-manager agent is deployed on various separate routers in the network. The following subsections describe the policies of the agents in monitoring and detecting the DDoS traffic.

MONITORING M-TRAFFIC

Users' traffic is first monitored at the ingress gateways where RED algorithm computes the AQS to detect incipient congestion at the gateway queues. Based on the congestion level, the burst gateways notify users of congestion by either marking a notification in their packet headers or dropping these packets. According to (Floyd and Jacobson, 1993), the possibility of notifying end-user by a particular burst gateway is almost commensurate with the user's share of bandwidth through that gateway. In this model, the Violation-monitor should be permanently active to filter traffic with RED notifications as *M-traffic*. However, traffic of users not notified by the RED notifications is *N-traffic*, which is normally allowed to be transmitted to the destination.

IDENTIFYING DDOS-TRAFFIC

A further investigation is performed on the *M-traffic* to verify if it is DDoS traffic by gathering PTR rates using the passive measurement technique. In this study, the attack is detected by identifying the intruders that strip others' resources. In the case of gateway burst, generating RED notifications activates the passive measurement phase to measure the PTR of *M-traffic*. The Violation-monitor agent probes the Violation-verifier agent to be activated and reports the total bandwidth consumed by each *M-user*. Thus, the PTR of each *M-user* is calculated and reported to the SLA-manager agent for comparisons with SLA_{PTR}. User that consumes PTR higher than the bandwidth share in the SLA is *intruder*. The SLA-manager agent then sends notification packets to all ingress edges to filter the traffic of that user as *DDoS-traffic*. Users within the bandwidth ratio guaranteed in the SLA are victims and their traffic is classified back to *N-traffic*.

EXPERIMENTAL RESULTS

This section presents the experimental results of the proposed model and evaluates its performance with recent existing schemes.

SIMULATION SETUP

Our experiment was conducted by using a network simulator NS-2.34 (ns-2). As shown in Fig. 3, the network topology is composed of 47 nodes (35 edge routers (E1–E35) and 12 core routers (CR1–CR12)). The network traffic can be generated by 70 end-users (U1–U70). Each user could use several active sources to send several flows through one or more ingress edges. Each of the 5 edges was gathered as a 5-fold set; likewise, each of the 10 end-users was gathered to obtain a denary set. Each end-user could send its data through one fivefold set of ingress edges as maximum, and each ingress edge could be used only by one denary set of end-users as maximum. Further descriptions of users' traffic setting are presented in Table 1.

Table 1. End-users traffic setting

End–users	Connected to	Destined to	Generated Traffic (Mbps/user)
U1-U10	E1-E5	E26	1
U11-U20	E6-E10	E20	0.7
U21-U30	E11-E15	E30	1
U31-U40	E16-E20	E10	0.5
U41-U50	E21-E25	E33	0.5
U51-U60	E26-E30	E1	0.7
U61-U70	E31-E35	E25	1

The maximum TCP flow window is 64 packets and the maximum packet size is 1024 bytes. QoS ratios guaranteed for end-users had been predefined by the SLA. Delay ratios had been specified according to the link delays. Jitter predefined ratios were 10% of the links delays. Bandwidth shares were equally specified for all users at 20% of core link bandwidth. Packet loss ratios were 1% of the PTR fraction of each user. The simulated experiment was run for 100 s.

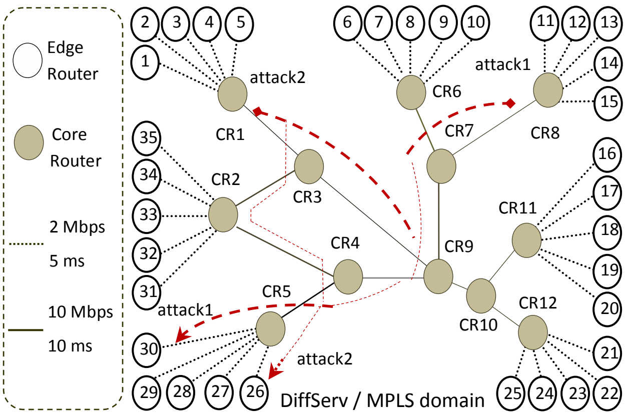

Figure 3. Simulated network topology

RESULT AND DISCUSSION

In this simulated experiment, network traffic was monitored under light load and traffic burst. Light load was observed at 0–9 and 86–100 s. Within these periods, end-users did not consume more than their bandwidth share; therefore, the network was properly provisioned. Traffic burst was observed at 10–85 s, when attacks were executed and the link bandwidth could no longer accommodate all user traffic. Within this period, two mutual attacks were simulated as DDoS traffic increased to more than 10 Mbps. Attack 1 was generated by U30 through ingress edges E11 to E15. Attack 2 was generated by U10 through ingress edges E1 to E5. Attacks 1 and Attack2 were intended for E30 and

E26, respectively; consequently, the edges of CR4 to CR5 became the most congested links. Figure 3 illustrates from which source edges the attack was injected and for which destination edges the attack was intended.

SCENARIO FOR MONITORING M-TRAFFIC

This scenario demonstrates the ability of the RED gateway for recognizing misbehaving user traffic. The values of RED parameters are set as follows: queue limit is 100 packets, min_{th} is 7 to10 packets, max_{th} is 21 to 30 packets, and w_q is 0.02. Figure 4 shows the values of AQS and QS in the domain gateways that were measured by using the RED algorithm. Gateways with an AQS between 7 and 21 packets marked the packet headers of misbehaving users with RED notifications. Based on the AQS curves, E11 to E15 started to stick the traffic of U20 to U30 with RED notifications at 16 s, and E1 to E5 started to stick the traffic of U1 to U10 with RED notifications at 32 s. Gateways with an AQS exceeding 21 packets notified *M-users* by dropping the excessive packets instead of marking them with RED notifications. E11 to E15 dropped the excessive packets from 58 s to 87 s. E1 to E5 dropped the excessive packets from 83 s to 91 s. During periods when gateway AQS was less than 7 packets, the network traffic did not experience dropping or RED marking. The RED notification was used in this experiment to identify *M-users* with traffic that requires further investigation. During the period of attacks, the number of *M-users* notified by the burst ingress edges with RED notifications oscillates to reach a maximum value of 20 *M-users*. Therefore, the RED-based strategy filters 20 *M-users* of 70 users in its worst case.

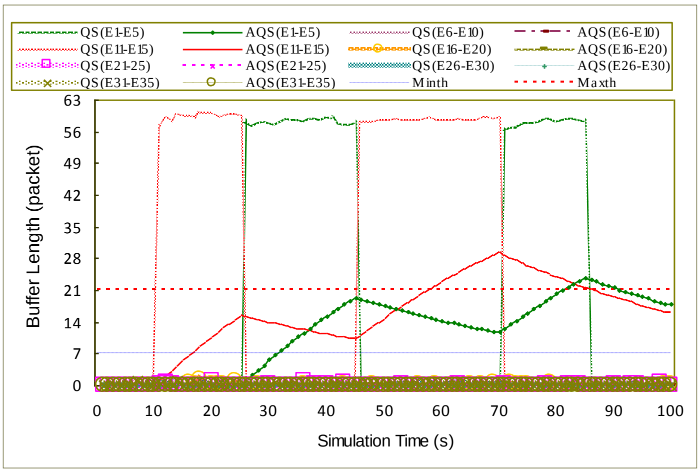

Figure 4. Domain RED gateways QS and AQS

SCENARIO FOR FILTERING DDOS-TRAFFIC

It is demonstrated that only users whose traffic classified as *M-user-traffic* violated the SLA. Hence, the PTR measurement was required only for traffic of 20 users, not 70. Using the passive technique, the PTRs of *M-users* are measured at the ingress edges. Figure 5 shows the link bandwidth percentage consumed by every *M-user*. U10 breached its SLA bandwidth guarantee by exceeding its bit rates and consuming more than 90% of the link bandwidth from 25 s to 45 s and from 70 s to 85 s. U30 exceeded its bit rate guarantee by approximately 90% from 10 s to 25 s and from 45 s to 70 s. However, the rest of the users did not exceed their PTR shares in the SLA, which was

20% of the intermediate link bandwidth. This scenario implies that U10 and U30 were intruders who launched the attack. Consequently, the excessive traffic sent by these intruders was filtered out as *DDoS-Traffic*. Conversely, U1 to U9 and U21 to U29 were victims plundered by the attacks of U10 and U30.

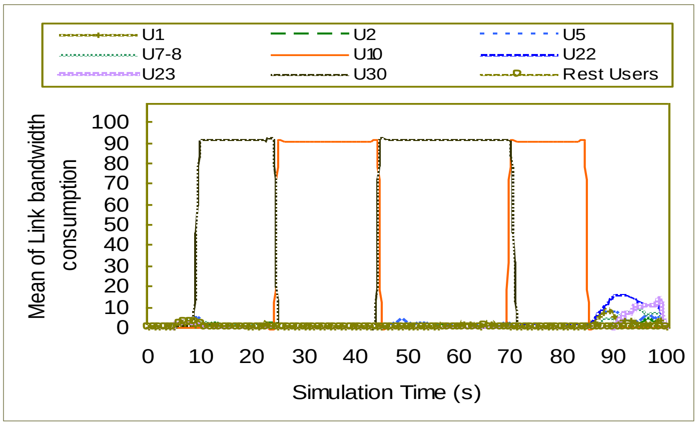

Figure 5. *M-users* consumption percentages of PTR.

In the current study, RED notification was an indicator used to identify *M-traffic*, which requires further investigation. Table 2 shows the *M-users* whose traffic had been classified as *M-traffic* in this scenario. The table also shows at what time and by which burst gateway the traffic of these users was filtered. For instance, E11 to E15 started to stick the traffic of U21 to U30 with RED notifications at the 16th s and stopped at the 57nd s. E11 to E15 started to stick RED notifications again to the same user traffic at the 88th s and continued until the end of the simulation time. E1 to E5 stuck U1 to U10 with RED notifications from 32 s to 82 s and from 92 s up to the end of the simulation time.

Table 2. Details of RED notifications issued for misbehaving users

Notified users	Generator edges	RED marking started at	RED marking
U21 to U30	E11 to E15	16th s	57th s
U21 to U30	E11 to E15	88th s	100th s
U1 to U10	E1 to E5	32nd s	82nd s
U1 to U10	E1 to E5	92nd s	100th s

ANALYSIS AND EVALUATION

The effective protection systems should be able to support high detection capability, a large-scale network in a scalable way, and should be reliable for monitoring whole network traffic. Thus, we evaluated the proposed investigation model (iModel for short) by comparing scalability with each Stripe-based (Habib, 2003) and Com-approach (Abdulghani, 2011) schemes. Figure 3 shows the topology used for the comparison analysis. Scalability evaluation was achieved by measuring processing overhead *POH* and communication overhead *COD* of each scheme with variable domain sizes. For computing the *COD*, the total number of probe packets injected per unit time for investigating network traffic was multiplied by the size of probe packets. However, the *POD* was computed by considering the extra processing P_{extra} at all hops *h*, through which a packet passes per unit time. For each probe packet in the monitoring schemes, a *POD* is required to change some fields in the packet header, such as address lookup, checksum computation, and any other CPU processing overhead. Thus, for a network domain with *U* users, and *N* edge routers, the *POH* and *COD* are calculated in each scheme.

In the Stripe-based approach, for each user, every edge injects a stripe of *S* packets to every egress edge pair. The egress edge pair sends a complementary stripe in reverse. Thus, *POD* and *COD* in the Stripe-based are computed by formulas (6) and (7), respectively.

$$Stripe_{POD} = (S \times (N-1) \times (N-2)) \times h \times P_{extra} \times U \tag{6}$$

$$Stripe_{COD} = (S \times (N-1) \times (N-2)) \times Pkt\_Size \times U \tag{7}$$

For monitoring the network in the Com-approach, every edge injects 4-packet trains *T* to every egress edge for each suspicious user *sU*. Thus, the *POD* and *COD* are given by (8) and (9), respectively.

$$Com-approach_{POD} = (T \times (N-1) \times (N-1)) \times h \times P_{extra} \times sU \tag{8}$$

$$Com-approach_{COD} = (T \times (N-1) \times (N-1)) \times Pkt\_Size \times sU \tag{9}$$

In the proposed iModel, any bursting edge probes edge routers to report their PTR fraction of the corresponding *M-users*. In response, the edge routers report the PTR fractions to the SLA-manager. Thus, the *POD* and *COD* of iModel are given by formulas (10) and (11), respectively:

$$iModel_{POD} = T \times (bN \times (N-1)) \times h \times P_{extra} \times mU \tag{10}$$

$$iModel_{COD} = T \times (bN \times (N-1)) \times Pkt\_Size \times mU \tag{11}$$

where *bN* is the bursting gateways and *mU* is the number of *M-users*. Hence, scalability of each scheme was evaluated by using variable number of gateways ranging from 35 to 2000, while the number of users was fixed at 70. The values of the parameters used in the formula are set as follows: *S* is 3 packets, *T* is 4 packets, *h* is 6 hops, *Pkt_Size* is 40 bytes, *bN* is 57% of *N*, and *mU* is 60% of *U*, where the values of *bN* and *mU* are computed based on the results demonstrated in Figure 4.

A comparison of the *POD* and *COD* among the three schemes showed that the iModel exhibits the lowest values, as shown in Fig. 7 and Fig 8. Additionally, evaluation of schemes' scalability with thousands of edge-routers demonstrated that iModel, when compared with the existing schemes, is able to support a large-scale network in a scalable way.

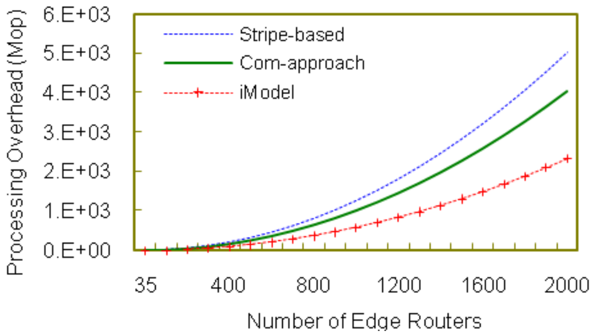

Figure 7. POD comparisons

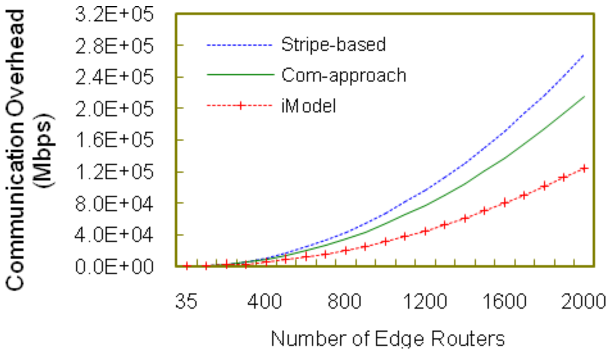

Figure 8. COD comparisons

CONCLUSION AND FUTURE WORK

The model presented in the current study detects the attack before it happens with early warning notifications to uncover the attackers while still in the planning stages of an attack. Monitoring the RED notifications as an early notification when anomaly congestion surfaces is beneficial for detecting DDoS traffic and minimizing potential overhead and resources associated with the attacks. Although the results are based on simulation scenarios, comparison of the approximate results indicates that the exploitation of RED notifications in triggering traffic investigation reduces more than 50% of investigation overhead and assists the proposed algorithm in supporting a large-scale network in a scalable manner.

Probing all ingress gateways to aggregate the bandwidth share of misbehaving users is also significant in identifying DDoS traffic, which could be carried out by multiple ingress edges in a capable manner. In the future, the behavior of the detected DDoS traffic can be analyzed to create DDoS traffic patterns. These patterns may be used in the preliminary phases of investigation to detect traffic with similar DDoS traffic behavior at an early time.

REFERENCES

Abdulghani, A. Ahmed, A. Jantan. G. Ahmed.(2010). A Potent Model for Unwanted Traffic Detection in QoS Network Domain"," JDCTA, vol. 4, pp. 122 ~ 130.

Abdulghani, A. Ahmed, A. Jantan, and T.C. Wan.(2011).SLA-based complementary approach for network intrusion detection," Computer Communications, vol. 34, pp1738-1749.

Ahsan, H. K. Maleq, and B. Bharat.(2004).Edge-to-edge measurement-based distributed network monitoring, Computer Networks." vol. 44, pp. 211-233.

An, J. S. P. G.(2006). Packet marking based cooperative attack response service for effectively handling suspicious traffic", LNCS, vol. 4318, pp. 182-195.

Choffnes, D. R.(2010). Service-level network event detection from edge systems, PhDdissertation, NORTHWESTERN university, p. 131.

Floyd, S. and V. Jacobson.(1993). Random early detection gateways for congestion avoidance, Networking, IEEE/ACM Transactions on, vol. 1, pp. 397-413.

Gu, Y. X. Hong, M. Mazzucco, and R. Grossman.(2003). Rate Based Congestion Control over High Bandwidth/Delay Links, IEEE/ACM Transaction on Networking, vol. 11.

Gyanchandani2, S. S. M.(2010). Analysis of Botnet Behavior Using Queuing Theory, IJCS, vol. 1, pp. 239-241.

Habib, A. S. Fahmy, S. R. Avasarala, V. Prabhakar, and B. Bhargava.(2003). On detecting service violations and bandwidth theft in QoS network domains, Computer Communications, vol. 26, pp. 861-871.

Habib, A. S. Fahmy, and B. Bhargava.(2005). Monitoring and controlling QoS network domains, International Journal of Network Management, vol. 15, pp. 11-29.

Jaeyeon, J. K. Balachander, and R. Michael.(2002). Flash crowds and denial of service attacks: characterization and implications for CDNs and web sites, in Proceedings of the 11th international conference on WWW, Hawaii, USA: ACM.

Jose, N. "DDoS attack evolution," Network Security, vol.(2008), pp. 7-10. Kulatunga, C. and G. Fairhurst.(2010). Enforcing layered multicast congestion control using ECN-nonce,Computer Networks, vol. 54, pp. 489-505.

Ningning, P. Hu Steenkiste.(2003). Evaluation and characterization of available bandwidth probing techniques Communications, IEEE Journal vol. 21, pp. 879 - 894.

Tham, C.-K. and Y. Liu.(2005). Assured end-to-end QoS through adaptive marking in multidomain differentiated services networks, Computer Communications, vol. 28, pp. 2009- 2019.

"The Network Simulator (ns-2) home page," http://www.isi.edu/nsnam/ns/.

Xuan I. S. Y., My T. Thai, Taieb Znati.(2010). Detecting Application Denial-of-Service attacks: A Group-Testing-Based Approach,IEEE Transactions on Parallel and Distributed Systems, vol. 21, pp. 1203-1216.