A Study of Traffic Survivability Under Malicious Attacks

Yen-Hung Hu*, Mira Yun†, Debra Tang‡ and Hyeong-Ah Choi†
*Department of Computer Science
Hampton University
Hampton, Virginia 23666
Email: yenhung.hu@hamptonu.edu
†Department of Computer Science
George Washington University
Washington, DC 20052
Email: {mirayun,hchoi}@gwu.edu
‡Software Engineering Center
US ARMY CECOM
Belvoir Fort, Belvoir, VA 22060
Email: debra.tang@us.army.mil

***Abstract*— As the Internet becomes more mature and a part of our daily life, the management of its resources to provide guaranteed services is crucial and beyond the capability of every individual network domain. How to maintain its continuous services in critical conditions is a challenge and would affect the design of next generation network infrastructure. In this paper, we study the survivability of normal traffics under the influences of flooding-based denial of service attacks and propose a potential framework which would reduce such influences. Our research increases the understanding of the behaviors of flooding-based DoS attacks and provides potential trends for developing better malicious attack mitigating solutions. By taking advantages of the survivability of specific flows, the potential mechanisms will not change or only slightly change current network infrastructure and be able to perform quality of service for several existing applications with no or little investment.**

I. Introduction

Providing minimum guaranteed service for normal flows coexisting with misbehaving malicious flows without downgrade of normal services is critical because of the vulnerabilities of software and hardware in network components and service providers, asymmetry of network resources between the Internet and victims, and distinct protocol responses to network congestion. In fact, a denial-of-service (DoS) attack [1] purporting to deny the services for normal flows could easily be achieved by exploiting system vulnerabilities, network resource asymmetry, and, especially, protocol biases. System vulnerabilities could be released by updating faulty components periodically or shortly after the weaknesses are explored. Resource asymmetry could be reduced by installing and updating resources. However the drawback of protocol biases has left a back door to potential risks and raised the instability of networks. Eventually, it has been widely used by DoS attacks and caused major losses in recent Internet security history [2].

Flooding-based DoS attack [3] is an easy way to cause service denial to normal traffics by exhausting resources of victims through high rate malicious flows. Many proposed approaches in literature for dealing with this attack are to provide fairness to all active flows or to drop suspicious malicious packets before they reach victims. Most of them have to modify infrastructure of networks on either network layer (e.g., queue policy, routing path, resource reservation) or protocol layer, and are not easy to be implemented when traffics travel across several different network domains [4], [5], [6], [7].

In this paper, we study the performance and survivability of normal and malicious flows under various network environments through large scale simulations. In particular, investigating a way that could improve the survivability of normal flows without modifying any network infrastructure is our main objective. To do this, we propose a novel framework which would reduce the influences from malicious flows.

The paper is organized as follows: Section II discusses the details of our simulated traffic and simulation setup. Section III shows our simulation results and observations. Section IV introduces the rationale of our framework. Summary and conclusions are presented in Section V.

II. Simulations

To study the survivability of normal flows, we have developed a simulated traffic (see Figure1) for NSF T1 network and NSF T3 network based on a report of Oregon Gigapop traffic [8]. The composition of this simulated traffic in terms of flow number, flow type (i.e., application), average link utilization, and total bytes for each protocol is very close to the information in [8]. In our simulations, the distribution of start time of flows is carefully designed to reduce the unbalanced load (i.e., burst) of links. Every flow, except Multicast, starts to transmit packets at its start time and stops when no packets are left or simulation is finished. Therefore, the stop time of

			flows #	flows %	interval/ms	pkt/flow	total packets	total bytes	bytes %
NSF T1	UDP	multicast	35	15.77	230	1307	45745	45745	93.40
		real	6	2.70		246	1476	1476	3.01
		half life	27	12.16		18	486	486	0.99
		icq	4	1.80		14	56	56	0.11
		dns	53	23.87		1	53	53	0.11
		others	98	44.14		12	1176	1176	2.40
	TCP	nntp	295	32.07		123	36285	39265	80.62
		ftp	15	1.63		81	1215	1215	2.70
		gnutella	33	3.59		30	990	990	2.20
		napster	10	1.09		97	970	970	2.16
		http	383	41.63		2	766	766	1.70
		kazaa	14	1.52		31	434	434	0.96
		others	174	18.91		25	4350	4350	9.66
NSF T3	UDP	multicast	1011	15.73	230	1307	1321377	132137	93.41
		real	172	2.68		246	42312	42312	2.99
		half life	775	12.05		18	13950	13950	0.99
		icq	103	1.60		14	1442	1442	0.10
		dns	1540	23.95		1	1540	1540	0.11
		others	2828	43.99		12	33936	33936	2.40
	TCP	nntp	8575	31.96		123	1055217	1055217	80.80
		ftp	416	1.55		81	33696	33696	2.58
		gnutella	955	3.56		30	28650	28650	2.19
		napster	280	1.04		97	27160	27160	2.08
		http	11155	41.58		2	22310	22310	1.71
		kazaa	401	1.49		31	12431	12431	0.95
		others	5060	18.85		25	126500	126500	9.69

Fig. 1. Simulated Traffic Pattern for NSF T1 and NSF T3.

each flow is determined by its flow size, packet number, rate, and network conditions. A Multicast traffic is treated as a long-life flow and always has packets waiting for delivering during entire simulation period. The average hop number of flows in NSF T1 and NSF T3 networks is 2.16 and 3.58 respectively.

A. Simulation Setup

In this paper, the bandwidth of links in NSF T1 is 1.544 Mbps and is 45 Mbps for links in NSF T3. Propagation delay of every link in both topologies is fixed to 10 ms. Two queue management algorithms are implemented: Drop-Tail and RED [9]. FIFO is the only queue scheduling algorithm used for these two queue management algorithms. The default values of four RED parameters used in our simulations are: minimum threshold is 5; maximum threshold is 15, maximum drop probability is 0.1, and weight factor is 0.002.

The malicious flows presented in our simulations are assumed to be active during the entire simulation period, and their sending rates are large enough to saturate links and course congestion. For example, the rate of a malicious flows injected into NSF T1 is 2 Mbps and is 60 Mbps for NSF T3. All of our simulations run from 0 seconds to 300.01 seconds, but traffics stop at 300 seconds. Throughput is calculated by counting total amount of bytes sent out and goodput is calculated by counting total amount of bytes successfully reaching the destination.

In order to create the congested network, we introduce a term *load factor* which is represented as (*new flow size* $\div$ *old flow size*). For example, *load factor* = 2 means the size of a flow is doubled when compared with its original size.

III. Performance Analysis

To study the survivability of normal flows, we investigate two cases: network without malicious flows, and network with malicious flows.

A. Network Without Malicious Flows

When there is no malicious flow, the bandwidth of a link is shared among the active flows passing through this link and bandwidth allocated for each active flow is based on several factors including queue management scheduling algorithm on routers, transportation protocol used, and flow size. Since more than 95% of traffic on the current Internet is composed by UDP and TCP protocols, in this paper the factors that affect the performance of UDP and TCP will be more emphasized than others.

Before we move on, we would like to examine several observations in [9]. The advantages of RED over DropTail are: (1) both TCP and UDP flows decrease end-to-end delays, (2) the loss of a large number of consecutive packets is prevented as it reserves some buffer spaces, and (3) the higher packet loss against bursty traffic is reduced. However, when a large scale simulation is performed as in our model, we find that some inconsistencies exist in some of the above observations. Our results in Figure 2 and Figure 3 show that when TCP flows are concerned, in some cases, DropTail provides better goodput as well as smaller end-to-end delays than RED. (See NNTP and FTP flows in these figures.)

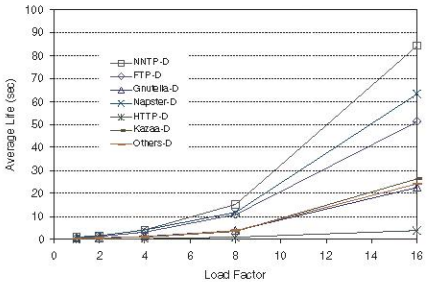

Fig. 2. NSF T1 topology, average life over TCP flows under DropTail.

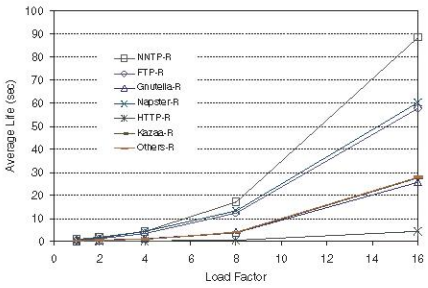

Fig. 3. NSF T1 topology, average life over TCP flows under RED.

B. Network With Malicious Flows

To study the influence of malicious flows coexisting with normal flow, we have injected high-rate UDP flows in NSF T1 and NSF T3 networks. Intuitively, normal UDP flows survive better than normal TCP flows since TCP flows will reduce their sending rates in response to the congestion caused by malicious flows while normal UDP flows continue to keep the same sending rates.

1) One Attack Model: Firstly, we consider a simple case. In this case, only one malicious flow is injected into networks. The source-destination pair of this malicious flow is arbitrarily selected and the hop number it traveled is varied. From our

simulation results, the average utilization of links affected by the malicious flow is decreased when the load factor of the normal traffic increases, i.e., the goodput of the malicious flow is relatively decreased, in both RED and DropTail. One interesting observation we have found is that the survivability of normal flows which do not travel through links affected by the malicious flow is in fact increased. Our simulation results confirm that RED performs better than DropTail since incoming packets start to drop before the buffer overflows. See Figure 4 and Figure 5 for detailed results.

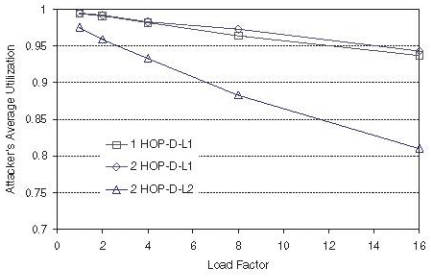

Fig. 4. NSF T1 topology, average utilization of malicious flow on the affected link. The notation X Hop-R-LY means that this is an average utilization of a link which is the Yth hops in the path traversed by the malicious flow which will traverse X hops in the network. DropTail algorithms is implemented.

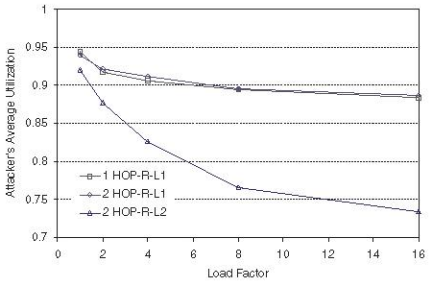

Fig. 5. NSF T1 topology, average utilization of malicious flow on the affected link. The notation X Hop-R-LY means that this is an average utilization of a link which is the Yth hops in the path traversed by the malicious flow which will traverse X hops in the network. RED algorithms is implemented.

2) Multiple Attack Models: We now consider multiple attack models. Two attacking models are implemented: *economic attack model:* in which approximately minimum number (20 malicious flows for NSF T1, and 25 malicious flows for NSF T3) of malicious flows are injected into the networks and each link is traversed by a malicious flow once; and *extreme attack model:* in which each malicious flow only affects one hop, and the number of malicious flows is the same as the number of links in the network. As shown in Figures 6, 7, 8, and 9, in the case of an extreme attack, RED provides better protection for normal traffic than DropTail. When RED is implemented, normal UDP can reserve 30% to 60% of goodput and almost keep constant when load factor increased (i.e., congestion increased), but such reservation will be at most 40% for most TCP traffic (except HTTP) and is getting worse when load factor increased. However, the simulation results for DropTail are worse than RED, in which only at most 35% of UDP flows and 20% of TCP flows (included HTTP) will be protected. We also found that the result for the economic attack shows that RED provides better protection for normal traffic as it was for the extreme attack.

We would like to point out one more interesting observation that the survivability of normal UDP flows is not directly related with the flow size (in RED, shorter UDP flows, e.g., ICQ, has better survivability than larger UDP flows, e.g., Real, but in DropTail it is not consistent). However, the survivability of TCP flows is directly related with the flow size such that the goodput is better for smaller size flows (e.g., HTTP) in both RED and DropTail.

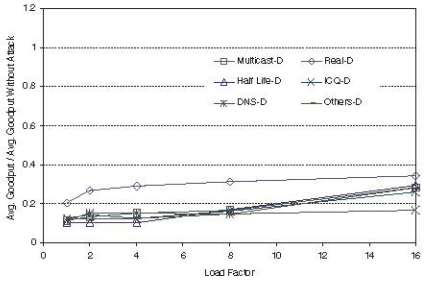

Fig. 6. NSF T1 topology, Extreme attacks under DropTail algorithm, *average flow goodput / average flow goodput without attacks* of UDP flows.

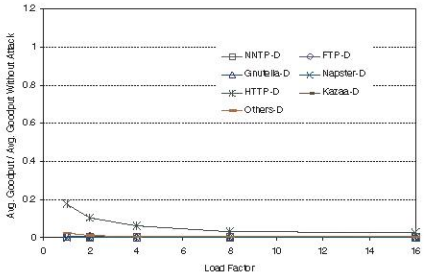

Fig. 7. NSF T1 topology, Extreme attacks under DropTail algorithm, *average flow goodput / average flow goodput without attacks* of TCP flows.

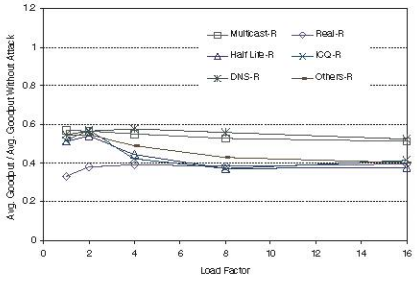

Fig. 8. NSF T1 topology, Extreme attacks under RED algorithm, *average flow goodput / average flow goodput without attacks* of UDP flows.

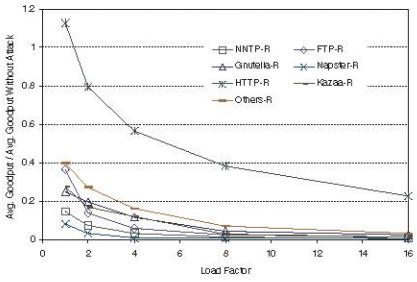

Fig. 9. NSF T1 topology, Extreme attacks under RED algorithm, *average flow goodput / average flow goodput without attacks* of TCP flows.

C. Observations

In this section, we have observed that: (1) When link utilization is very high (i.e., the network is congested and many packets are dropped) RED will provide better performance than DropTail. Otherwise DropTail has better performance. (2) No matter which queue management algorithm is adopted, UDP traffic will have better survivability than TCP. (3) Delay of UDP traffic will be smaller than TCP, when they have the same flow size and packet number and as long as packet loss of UDP traffic is carefully handled (need a mechanism for retransmitting lost packets). (4) Smaller flow size will provide better survivability than larger flow size when RED is implemented (Even, in DropTail, the result of UDP traffic does not be apparently supported. However we believe when the UDP retransmitting mechanism in an application layer is implemented, the behaviors of UDP flows will be close to those of TCP flows, and this observation will hold).

IV. Malicious Attack Mitigating Framework

It is not easy to modify any network infrastructure (queue algorithm, protocol, etc.) without the cooperation of the majority of users and manufacturers. Therefore, with the help from the observations gathered from our simulation results, we propose a framework (see Figure 10) which could provide better survivability and QoS for normal application flows but without performing any modification of current network infrastructure. The main ideas of our framework are:

- When network is congested, RED will be selected as the queue management algorithm on routers. Otherwise, DropTail is the default choice. In the framework, queue management algorithm (RED or DropTail) is dynamically chosen according to network conditions.
- Since having shown better performance than TCP protocol, UDP protocol will be the candidate for our framework. However, because UDP protocol does not provide guaranteed delivery for every packet, an application layer packet loss retransmitting mechanism has to be implemented in both ends to enhance the performance of UDP protocol.
- To increase the survivability of normal flows, instead of using only one connection for each flow, multiple connections with smaller flow size for each will be considered (i.e., if flow size is L and N connections are made for this flow, then the flow size of each connection will be $\frac{L}{N}$).

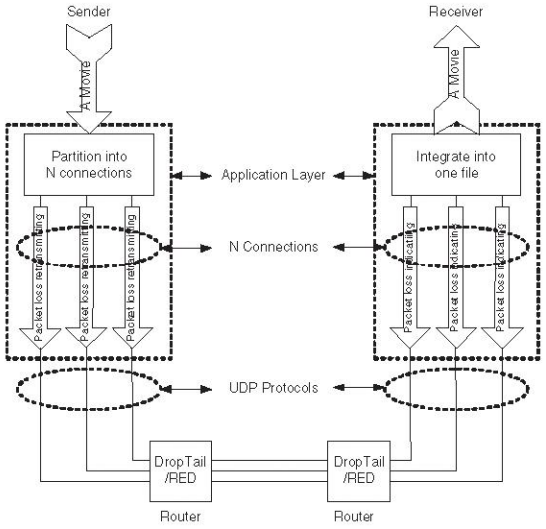

Fig. 10. Malicious Attack Mitigating Framework

V. Conclusions

In this paper, we studied the performance of normal and malicious flows under various network environment through large scale simulations. Our simulation study showed that (1) when the network is congested RED will provide better performance than DropTail, (2) UDP flows will have better survivability than TCP under RED and DropTail, (3) delay of a UDP flow would be smaller than a TCP flow, (4) smaller flow size will provide better survivability than larger flow size when RED is implemented. Finally, based on the observations gathered from our simulation results, a novel framework is proposed, which gives an idea of mitigating the effects of the malicious attacks. The presented results are believed to be useful in developing control mechanisms counteracting network congestion caused by flooding-based malicious flows.

References

[1] R. K. C. Chang, *"Defending against flooding-based distributed denial-of-service attacks: a tutorial,"* IEEE Comm. Magazine, Oct. 2002.
[2] *"COMPUTER CRIME AND SECURITY SURVEY,"* CSI/FBI, 2003.
[3] A. Piskozub, *"Denail of service and distributed denial of service attacks,"* in Proc of TCSET'2002, Feb. 18-23, 2002, Lviv-Slavsko, Ukraine.
[4] I. Stoica, S. Shenker, and H. Zhang, *"Core-stateless fair queueing: a scalable architecture to approximate fair bandwidth allocations in high-speed networks,"* IEEE/ACM Trans. Networking, vol. 11, no. 1, Feb. 2003.
[5] J. S. Li and M. S. Leu, *"Fair bandwidth share using flow number estimation,"* in Proc. of IEEE ICC 2002, New York City, USA.
[6] Yen-Hung Hu, Hongsik Choi, Hyeong-Ah Choi, *"Packet Filtering for Congestion Control under DoS Attacks,"* in proc. of IWIA 2004: 3-18
[7] Yen-Hung Hu, Hongsik Choi, Hyeong-Ah Choi, *"Packet Filtering to Defend Flooding-Based DDoS Attacks,"* in proc. of IEEE Sarnoff 2004.
[8] Joe St Sauver, *"Oregon Gigapop Traffic Characteriztion,"* Internet2/NLANR Joint Techs, May 16th 2001, Lincoln NE.
[9] S. Floyd, V. Jacobson, *"Random Early Detection Gateways for Congestion Avoidance,"* ACM Transactions on Networking, pp.397-413, Aug. 1993.