A Round-Trip Time-Based Prevention Technique to Secure LEO Satellite Networks from Denial-of-Service Attacks

Tarik Taleb, Nei Kato, and Yoshiaki Nemoto
Graduate School of Information Sciences - Tohoku University
Sendai, Japan 980–8579
Email: {taleb,kato,nemoto}@nemoto.ecei.tohoku.ac.jp

Abstract— **The development of satellite networks has recently gained a tremendous interest. The main reason beneath this interest underlies in the vision of anywhere, anytime pervasive access to the Internet over satellites.**

Protection of satellite systems from Denial-of-Service (DoS) attacks, a serious security threat in today's Internet, is a one major step towards making this vision a reality. The paper proposes a method to detect DoS attacks in the vicinity of flooding sources and in early stages before they cripple the system. The fundamental challenge in attack detection consists in distinguishing between simple flash events and DoS attacks so as not to deprive innocent users from having legitimate accesses. In the proposed mechanism, this distinction is based on the fact that legitimate TCP flows obey the congestion control protocol, whereas misbehaving sources remain unresponsive.

Suspicious flows are sent a test feedback and are required to decrease their sending rates. Legitimacy of such flows is decided based on their responsiveness. The scheme performance is evaluated through a set of simulations and encouraging results are obtained: short detection latency and high detection accuracy.

I. Introduction

Because of their extensive geographic reach and flexible deployment features, satellite network systems are seen as an attractive solution to realize the vision of global personal communications. Their design and development have been, thus, the subject of extensive research in recent literature and have gained a tremendous interest even at commercial level. Satellite systems can be classified into two kinds: Geostationary (GEO) and Low Earth Orbit (LEO) satellite systems. Despite the wide commercial usage of GEO systems in the last two decades, requirements for lower propagation delays, in conjunction with coverage of high latitude regions, have turned the light on LEO systems.

Current LEO satellite systems, such as Iridium, have generally been designed for voice-only communication. However, the recent financial failure of the Iridium system has made researchers realize that systems optimized for only voice or low data rates services may turn out unfavorable. The success of next-generation LEO satellite networks hinges, thus, on their ability to provide broadband data rates applications; similar in spirit to today's Internet.

Recent events have illustrated that the underlying Internet infrastructure is exposed to a high risk of denial-of-service (DoS) attacks. Although the full impact of DoS attacks on the security of LEO satellite network systems is yet to be felt, the seeds of these security concerns need to be considered if LEO systems are to be envisaged as providers of applications similar to the current Internet's. Without adequate security, cyber terrorism may deprive large enterprises from making efficient use of satellite systems and defense organizations may fail to guarantee the safety of their personnel in battlefield scenarios. To make the vision of the Internet over satellite networks a reality, the vulnerability of LEO satellite network systems to DoS attacks has to be discussed. This challenging task underpins the research work outlined in this paper.

The remainder of this paper is structured as follows. Section II highlights the relevance of this work to the state-of-art in the context of DoS attacks detection techniques. Section III presents the types of DoS attacks envisaged in this paper. The key design philosophy and distinct features that were incorporated in the proposed scheme are described in Section IV. Section V portrays the simulation environment and reports the simulation results. Following this, the paper concludes in Section VI with a summary recapping the main advantages and achievements of the proposed scheme.

II. Related Work

In the recent literature, several approaches have been proposed to counter DoS attacks. They can be classified into two types: traceback and prevention techniques. The former commence their search for attackers upon the collapse of a victim system, or a sharp degradation in performance. Most DoS traceback techniques are based on comparisons among traffic patterns or on packet marking. Whilst these techniques may be efficient in terrestrial networks, they may run into difficulty in the case of satellite networks. The main reason behind this limitation lies beneath the motion characteristic of both the satellite network and end-users. Consider a scenario where a DoS attack source is roaming over coverage areas of different satellites while flooding a victim with malicious traffic (Fig. 1). In such a scenario, applying traceback techniques to pin the real attacker down would result in unsuccessful monitoring of traffic coming from the traversed satellites, which ultimately would lead to confusing results. The traceback results would be even more ambiguous when handover occurs among satellites.

Prevention techniques, on the other hand, attempts to throttle attacks before they severely harm the system. The main strategy used in these techniques is ingress filtering. Recently proposed prevention techniques rely on monitoring changes in

0-7803-8521-7/04/$20.00 © 2004 IEEE

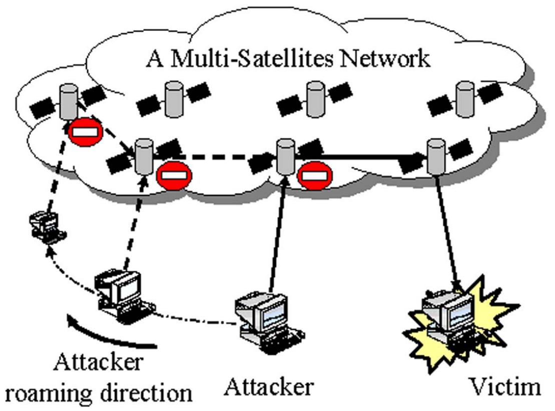


Fig. 1. Inefficiency of traceback techniques in satellite networks

the internal characteristics of the network, such as the traffic volume, loss ratio, and queuing delays. Other prevention techniques watch for the behavior of specific packet types to detect DoS attacks and alert the victim. [1] is a notable example. Upon detection of a change in network characteristics, most prevention techniques take the harsh measure of shutting off the traffic destined to the victim. Such a draconian measure can be seen as unfair towards some legitimate packets that may be contained in the blocked traffic. In satellite systems, given the fact that a single satellite has an extensive coverage area, such unfair event may easily occur when a potential number of legitimate users, from the same coverage area, access the same server simultaneously. Having multiple users accessing the same server at nearly the same time is referred to as "flash crowd" event throughout this paper. The challenge in this research is how to distinguish between traffic increase due to DoS attacks and that due to flash crowd events.

III. ENVISAGED TYPES OF DOS ATTACKS

Based on a backscatter analysis, [2] has indicated that over 94% of DoS attacks use TCP packets, comprising the remaining 6% UDP and ICMP packets. It has been shown also that a potential number of networks were victims of DoS attacks and had their vital links overloaded with unnecessary traffic. Knowing that flooding packets destined to a notorious UDP port can be easily identified as a DoS attack, this paper focuses on thwarting TCP-based attacks that attempt to overload servers or networks with useless traffic. The paper excludes the case of TCP-based attacks that consist of multiple TCP connections with less than three packets. The anomaly or legitimacy of such flows can be easily judged and there is a substantial set of mechanisms to cope with such ill-behaved connections in the recent literature [1].

Another issue is related to Distributed-DoS (DDoS) attacks. In terrestrial networks, a large-scale distribution of a DoS attack may make detection accuracy poor in the vicinity of the flooding sources. In satellite networks, however, DDoS attacks do not represent a major issue. The aggregate traffic of individual flooding traffic coming from different attacking sources within the coverage area of a single satellite can be considered as a single DoS attack.

To analytically demonstrate this observation, let $\aleph$ be the number of satellites in a given satellite constellation. Let $\Re$ and Ψ denote the minimum aggregate flooding rate of packets per second that should be sent to cripple a server and the number of coverage areas over which the attack is distributed, respectively. Considering the fact that the ocean represents 70% of the earth's surface and that it is hard to launch attacks from oceans due to the access limit, Ψ should be intuitively smaller than $(0.3 \cdot \aleph)$. On the other hand, [2] has reported that a rate of at least $15000 pkts/s$ is needed to attack a protected server ($\Re = 15000\ pkts/s$). Therefore, the most highly-distributed DoS attack should send attacking packets at individual rates of $(\frac{50000}{\aleph}\ pkts/s)$. In the case of the Iridium system, this minimum individual rate equals $757\ pkts/s$; a rate large enough to be detected by the proposed scheme, as is confirmed by simulation results. It should be recognized also that the above analysis represents clearly the worst-case scenario where the DoS attack is distributed globally, a fact that is highly unlikely to happen due to the access limit over even the land surface.

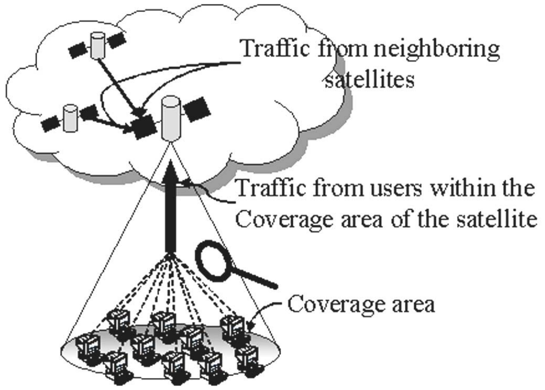

Traffic from neighboring satellites
Traffic from users within the Coverage area of the satellite
Coverage area

Fig. 2. Only the traffic issued from users within the coverage area of the satellite in question that is subject to monitoring; traffic coming from other satellites is not taken into consideration

IV. RTT-BASED DOS ATTACK PREVENTION SCHEME

This section gives a detailed description of the proposed attack detection method. Detection is performed at first-edge satellites without any involvement of core satellites. Only the traffic issued from users within the coverage area of the satellite in question that is subject to monitoring; traffic coming from neighboring satellites is not taken into consideration (Fig. 2). Throughout this paper, throughput refers to the total bandwidth consumed by traffic coming from end-users within the coverage area of the considered satellite. The throughput is computed over a detection resolution, Δ time interval. The choice of Δ is a compromise between the detection latency and the required computational load. In deed, small values of Δ would enable the detection engine to quickly detect an attack, whereas large values of Δ would reduce the computational load as the system checks the traffic less often.

For each satellite, three running states are defined: normal, alert and action. Under normal conditions, the edge satellite resides in normal state, watching for abnormal traffic behavior. When the increase in bandwidth consumption[1], $\partial_{\Re}$, exceeds a

[1] In this paper, the monitoring procedure is based on only the bandwidth consumption. History of the loss rate and queuing delay may, however, be used as detection features to improve further the detection accuracy.

pre-defined threshold, Θ,

$$\partial_{\Re} > \Theta \tag{1}$$

the monitor considers it a possible DoS attack. Throughout this paper, the parameter θ is set to

$$\Theta = \frac{50000}{\aleph} \; pkts/s \tag{2}$$

where $\aleph$ is the number of satellites in the considered constellation.

A high variation in bandwidth consumption is an indication of abnormal behavior inside the coverage area of the satellite. This argument is based on the fact that a DoS attack should inject a significant amount of traffic into the satellite to clog the targeted victim. Upon a noteworthy variation in bandwidth consumption, the satellite switches to alert state. In the alert state, the satellite clusters flows coming from users within its coverage area into a number of groups. Flows are defined as streams of packets sharing the quintuple: source and destination addresses, source and destination port numbers, and protocol field. Clustering of flows can be performed according to different elements. IP source and destination addresses are useful in forming aggregates of requests that are issued to access a particular server. Application type can be considered in case of a virulent worm that propagates by email and is overwhelming other traffic. IP destination prefix can be used in case of detection of flooding attacks targeting a site or a particular network link.

This paper considers the most common case; DoS attacks targeting a particular host or a set of web servers within the same domain (the same coverage area). Clustering is performed, thus, according to the IP destination prefix. Once the clustering procedure is done, the satellite sorts the clusters according to their aggregate traffic rate. For the cluster with the highest rate[2], the satellite first checks for any possible spoofed addresses. Filtering by ingress router at the satellite can verify the identity of coming packets and can detect IP address spoofing if the attacking packets use spoofed source addresses from outside the coverage area of the satellite [3]. However, the shortfall of this filtering technique is that if the IP addresses of the attacking packets are from the prefix range of the coverage area of the satellite, it does not always detect address spoofing. Retrieval of spoofed addresses can be used as a strong indication to increase the level of attack likelihood and to stimulate the satellite to enter the action state.

Having a high-rate cluster of numerous flows addressed to the same IP destination prefix, the system can infer that this is either an ordinary flash crowd event or a DoS attack. The challenge consists in making distinctions between the two cases. In the proposed mechanism, this distinction is based on the fact that legitimate TCP flows obey the Additive Increase Multiplicative Decrease (AIMD) concept of the congestion control protocol, whereas misbehaving sources remain unresponsive.

Before delving into details of the proposed DoS detection scheme, first is a description of how the scheme exploits some features of satellite networks to make an approximate estimate of flows RTTs and their legitimate sending rates. Prior knowledge of RTT estimates is usually not available at network elements in terrestrial networks. However, in LEO constellations, flows RTTs can be handled by a simple monitoring of hops count in the backward and forward traffic of each flow [4]. The hops count of each flow can be easily computed from Time to Live (TTL) field in the IP header of both ACK and data packets. Let H_b and H_f denote the number of hops traversed by an acknowledgment packet in the backward traffic and the number of hops traversed by a data packet in the forward traffic before entering the satellite in question, respectively. Since queuing delays make a minimal contribution in the one-way propagation delay and the value of the inter-satellite link delay remains constant, the estimated value of the connection RTT can be expressed as

$$RTT = 2 \cdot (H_b + H_f + 2) \cdot ISL_{delay} \tag{3}$$

where ISL_{delay} denotes the inter-satellite link delay.

Applying the mathematical model developed in [5] and using the RTT knowledge, the satellite can make an estimate of the correct sending rate, λ, of flows belonging to the suspect cluster. Note that flows from the same cluster are destined to the same area. They thus traverse the same number of hops, and consequently have the same RTT and similar sending rates. From the monitoring results of the cluster, the satellite can compute the actual average rate of the flows. Let μ_i denote the measured rate of the i^{th} flow in the cluster.

In the case of misbehaving flows with measured rates greater than $(1+\alpha)$ times the estimated rate λ

$$\mu_i > (1+\alpha)\lambda \tag{4}$$

the satellite sends them a test feedback requiring them to decrease their sending rates to a particular value. α points to the flow blocking tolerance of the scheme. If a flow does not react in a single RTT then it is unresponsive and its packets are discarded. Since flows are unaware of when the satellite is monitoring their behavior, they have to always follow the test feedback. The test feedback is written in the receiver's advertised window (RWND) field carried by the TCP header of acknowledgment packets. When the feedback reaches the sender, a legitimate user should react to the message and accordingly modify its current rate. It should be emphasized that during the monitoring process, TCP flows that have no ACK packets can be considered as part of a DDoS attack.

Transmission of the test feedback can be performed in a deterministic or probabilistic manner. In the deterministic case, all flows in the suspect cluster are sent the test feedback. The obvious drawback of the deterministic manner is that it incurs a significant processing overhead. In case of a dense cluster, transmission of the test feedback can be performed in a probabilistic manner. The pitfall of the probabilistic approach is that if the selected flows turn out to be unresponsive, all the flows of the cluster with rates satisfying equation (4) would be

[2] Note that this rate should be larger or equal to $\partial_{\Re}$

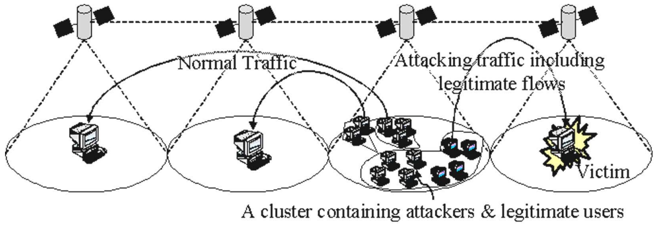

Normal Traffic
Attacking traffic including legitimate flows
Victim
A cluster containing attackers & legitimate users

Fig. 3. Simulation environment

blocked and may consequently frustrate some legitimate flows that may exist in the cluster. Finally, it should be notified that by implementing the proposed method as a background task, the processing power required to identify suspect clusters and to send their flows a test feedback should not be an issue.

V. Performance Evaluation

Having described the details of the proposed scheme, focus is now directed on its performance evaluation. This section verifies how the proposed system is efficient in protecting satellite networks from DoS attacks while causing no damage to innocent flows. The performance evaluation relies on computer simulation, using Network Simulator (NS) [6]. Therefore, particular attention is paid to the design of an accurate and realistic simulation setup, which is described below, justifying the choices made along the way.

Fig. 3 depicts the used network configuration. The figure considers the case of a satellite serving a population of N terminals. The N users form M clusters based on the prefix of their IP destination addresses. For the sake of simplicity, the DoS attack is assumed to target a single victim located in the coverage area of a different satellite, as shown in the figure. Obviously, this assumption has no effect on the overall performance of the proposed detection method. While $(M-1)$ clusters represent no danger to the network and contain only legitimate flows that send data to destinations other than the victim, one cluster is assumed to contain flooding sources as well as legitimate ones. Let N_s denote the size of this suspicious cluster. Throughout this paper, χ denotes the percentage of attacking sources among the flows of the suspicious cluster. In the simulation, χ is varied to model different levels of aggressiveness of the attacking aggregate. It is assumed that the satellite under study is traversed by only packets coming from users within its coverage area, whereas traffic from neighboring satellites is ignored. This assumption is made with no specific purpose in mind and does not change any of the fundamental observations about the simulation results. In all simulations, the inter-satellite link delay is set to 20 ms and all links, including up-links and down-links, are given capacities equal to 100 $Mbps$.

Legitimate users implement the TCP NewReno version [7]. The DoS attack is modeled as several ON/OFF UDP sources whose On/Off periods are of equal times and are chosen randomly between 2 s and 10 s. Each attacking flow sends malicious packets at a rate derived from a uniform distribution with a mean μ_{mean} and a variance of σ^2. In NS implementation, the maximum and minimum values of the distribution are set to $(\mu_{mean}+\sigma)$ and $(\mu_{mean}-\sigma)$, respectively ($max_{-}=\mu_{mean}+\sigma$, $min_{-}=\mu_{mean}-\sigma$). At the beginning of the simulation, we start the legitimate flows and let them stabilize. At time $t=5$ s, the flooding sources are activated. Simulations are all run for 30 s, a duration long enough to ensure that the system has reached a consistent behavior. The data packet size is fixed to 1 kB. All results are an average of 7 simulation runs. Table I shows a complete list of the simulation parameters and the range of values studied.

TABLE I

Simulation parameters and range of values

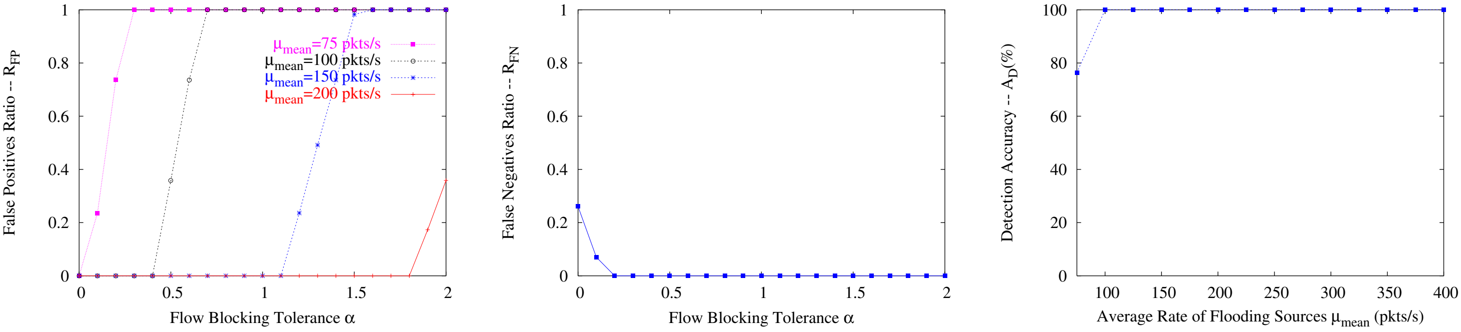
Factor	Simulation Parameters and Range of Values
Total number of flows N	15000
Number of clusters M	3
Size of the suspicious cluster N_s	1000 - 10000
Attack aggressiveness χ	0.3 - 1
On/Off Periods of UDP flows	2 s – 10 s
Average rate of UDP flows μ_{mean}	70 pkts/s - 400 pkts
Variance of UDP flows rates σ	$0.17 \cdot \mu_{mean}$
Detection resolution Δ	300 ms - 2 s
Bandwidth consumption threshold Θ	757 pkts/s
Flow blocking tolerance α	0 - 2

Four quantifying parameters are used to evaluate the performance of the proposed scheme:

- False positives ratio (R_{FP}): This measure involves the number of malicious flows that go undetected by the system (False Positives N_{FP}). It should be always maintained in the vicinity of zero. R_{FP} is defined as
$$R_{FP}=\frac{N_{FP}}{\chi\cdot N_s}$$
- False negatives ratio (R_{FN}): This measure is defined as the ratio of innocent flows that are unfairly punished (False Negatives N_{FN}) to the total number of legitimate flows. This index should be always minimized to zero.
$$R_{FN}=\frac{N_{FN}}{(1-\chi)\cdot N_s}$$
- Detection accuracy (A_D): This measure captures the overall performance of the system and is defined as
$$A_D=(1-R_{FP})\cdot(1-R_{FN})\cdot 100$$
- Detection latency: The attack detection time is computed as the sum of the detection delay from the start of the attack till the detection of the first malicious flow and the time required to judge whether the flow was responsive to the submitted test feedback or not[3]. Obviously, the detection time should be as short as possible to allow the system to detect attacks soon and to accordingly minimize or eliminate the attack damage.

Fig. 4(a) graphs the false positives ratio for different values of α. For all the considered rates of the flooding sources μ_{mean}, simulation results show that higher values of α lead to higher number of false positives. This is due to the fact that in case of high values of α, some malicious flows went

[3] This operation was not assumed in the simulation as we modeled flooding sources as several UDP sources. This delay was thus simply set to a single RTT in the computation of the detection latency.



(a) False positives ratio vs Detection tolerance ($\chi = 0.7$, $N_s = 1000$, Deterministic method)

(b) False negatives ratio vs Detection tolerance ($\chi = 0.5$, $N_s = 10000$, $\mu_{mean} = 100 pkts/s$, Probabilistic method)

(c) Detection accuracy in case of dynamic setting of α ($\chi = 0.5$, $N_s = 10000$, Probabilistic method)

Fig. 4. Overall performance in terms of detection accuracy

undetected as their sending rates did not satisfy equation (4). On the other hand, Fig. 4(b) shows that smaller values of α caused the blocking of some legitimate users which have led eventually to higher values in the false negatives ratio. These false negatives are mainly due to the slight error that likely happened in the estimation of both flows RTT and their correct sending rate λ. The results show that the system performance can be critically affected by the choice of the attack tolerance α and that a tradeoff between the two quantifying parameters R_{FN} and R_{FP} should be established. It should be noted that guaranteeing smaller rates of false negatives is worthwhile even at the cost of a slight increase in the false positives. In deed, by blocking a set of flooding attacks, the undetected flooding sources (the false positives) have to significantly increase their flooding rates to bring down the victim under protection. This increased flooding traffic makes it easier to detect the flooding attack and block again its sources by the proposed system.

Being interested in minimizing the influence of α on the system performance, we set α in a dynamic way according to the following method

$$\alpha \quad = \quad \frac{\sqrt{\sum_i \left(\frac{\mu_i}{\lambda} - 1\right)^2}}{n} \qquad \forall i \quad \mu_i > \lambda \qquad (5)$$

where n is the number of flows in the suspicious cluster that have rates larger than λ.

Fig. 4(c) shows the detection accuracy of the system in case of setting α in a dynamic manner for different average rates of malicious flows. Excluding the case of attacks that consist of multiple connections with significantly small sending rates[4], the obtained results show that the detection accuracy of the proposed system is always high. As for the detection latency, since the system uses the explicit feedback to test a source, isolating misbehaving sources is fast and easy. In all simulations, the detection latency was smaller than $(\Delta + 3 \cdot RTT)$, where RTT is the estimated RTT of flows that belong to the suspicious cluster.

VI. Conclusion

In this paper, we proposed a RTT-based prevention technique to throttle TCP-based bandwidth attacks over satellite networks. The proposed method has the potential of distinguishing between ordinary flash events and DoS attacks. It has also the advantage of detecting DoS attacks in early stages before they severely harm the victim or the network links. The proposed method uses an explicit feedback to test sources and judges accordingly their legitimacy. In deed, suspicious flows are sent a test feedback and are required to decrease their sending rates. Unresponsive flows are blocked. Performance evaluation relied on computer simulation and a set of scenarios was considered. The obtained results were encouraging and elucidated both the short latency and high accuracy of the detection scheme.

Finally, it should be noted that the proposed system considers a sudden increase in the traffic volume as an indication of a bandwidth attack. Being aware of the possibility of being easily misled by this assumption due to the bursty nature of the Internet traffic, the authors are currently investigating an auto-regressive method to model the Internet traffic and to infer a possible attack based on its results.

References

[1] H. Wang, D. Zhang, and K.G. Shin, *"Detecting SYN Flooding Attacks"* In Proc. of IEEE Infocom, NewYork, June 2002.

[2] D. Moore, G.M. Voelker, and S. Savage, *"Inferring Internet Denial-of-Service Activity"*, In Proc. USENIX Security Symposium, Washington D.C, Aug. 2001.

[3] P. Ferguson and D. Senie, *"Network Ingress Filtering: Defeating Denial of Service Attacks which Employ IP Source Address Spoofing Agreements Performance Monitoring"*, Network Working Group, RFC 2827, May 2000.

[4] T. Taleb, N. Kato, and Y. Nemoto, *"An Explicit and Fair Window Adjustment Method to Enhance TCP Efficiency and Fairness over Multi-Hops Satellite Networks"*, IEEE J. Select. Areas Commun., vol. 22, no. 2, pp.371-387, Feb. 2004.

[5] J. Padhey, V. Firoiu, D. Towsley, and J. Kurose, *"Modeling TCP Throughput: A simple Model and its Empirical Validation"*, in Proc. of ACM SIGCOMM, Vancouver, B.C., Sep. 1998.

[6] UCB/LBNL/VINT, *"Network Simulator - ns (version 2)"*, http://www.isi.edu/nsnam/ns/

[7] S. Floyd and T. Henderson, *"The NewReno Modifications to TCP's Fast Recovery Algorithm"* Network Working Group, RFC 2582, Apr. 1999.

[4] Recall that there is a substantial set of mechanisms to cope with such attacks in the recent literature.