DoS protection for a Pragmatic Multiservice Network Based on Programmable Networks[1]

Bernardo Alarcos[1], María Calderón[2], Marifeli Sedano[3], Juan R. Velasco[1]

[1] Department of Automática, Universidad de Alcalá, Madrid, Spain.
{bernardo, juanra}@aut.uah.es
[2] Department of Ingeniería Telemática, Universidad Carlos III de Madrid, Madrid
maria@it.u3cm.es
[3] Department of Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid, Madrid, Spain
marifeli@gsi.dit.upm.es

Abstract. We propose a scenario of a multiservice network, based on pragmatic ideas of programmable networks. Active routers are capable of processing both active and legacy packets. This scenario is vulnerable to a Denial of Service attack, which consists in inserting false legacy packets into active routers. We propose a mechanism for detecting the injection of fake legacy packets into active routers. This mechanism consists in exchanging accounting information on the traffic between neighboring active routers. The exchange of accounting information must be carried out in a secure way using secure active packets. The proposed mechanism is sensitive to the loss of packets. To deal with this problem some improvements in the mechanism has been proposed. An important issue is the procedure for discharging packets when an attack has been detected. We propose an easy and efficient mechanism that would be improved in future work.

Key words: Active Networks, Security, Denial of Service

1 Introduction

Active and programmable networks [1] facilitate the provision of new dynamic services, introducing programmability into some nodes. We propose the use of an active router based on the SARA[2] platform [2] to build a pragmatic multiservice network.

The users of the multiservice network can request services (e.g. caching, transcoding of multimedia flow…) to improve the communications between the end system users and other end systems in the network. To offer a service, active routers execute some specific codes to process packets exchanged between end systems. The multiservice network requires security services to guarantee that only authorized users can deploy services.

[1] This work has been funded by CICYT under project IMPROVISA (TSI2005-07384-C03)

[2] SARA (Simple **Active Router**-Assistant Architecture) is a prototype of an active router developed under the GCAP IST project by the active networks researchers group of the Carlos III University of Madrid

Active routers are the critical point that must be protected. The active routers execute dynamic codes to process active packets carrying control information and legacy packets carrying data. The active packets must be authenticated to avoid fake active packets changing the behaviour of active routers. We have proposed a security solution to tackle these problems in [3]. Other problem can appear when fake legacy packets, which are also processed by active routers, are injected into active routers. A Denial of Service (DoS) attack occurs in active routers in this situation. In this paper we describe a solution to tackle this problem, which consists in exchanging accounting information between the neighbouring active routers. The insertion attacks can be detected at the attacked nodes by comparing the exchanged information. A reaction mechanism to use when an attack occurs is also defined.

The rest of the paper is as follows: in section 2 we describe a proposal for the multiservice network, in section 3 the security problem in this scenario is presented, in section 4 we propose a mechanism to solve the problem, in section 5 some validation tests are presented, and finally section 6 is devoted to the conclusion and future work.

2 A Pragmatic Vision of Multiservice Networks Based on Programmable Networks

The multiservice network that we propose is made up of a number of active routers within an IP network. The active routers identify special packets called active packets and load a specific code to process these active packets. Active packets go from an end system (source) to an end system (destination), and the active routers in the path between the source and the destination process the active packets (Figure 1) using a specific code.

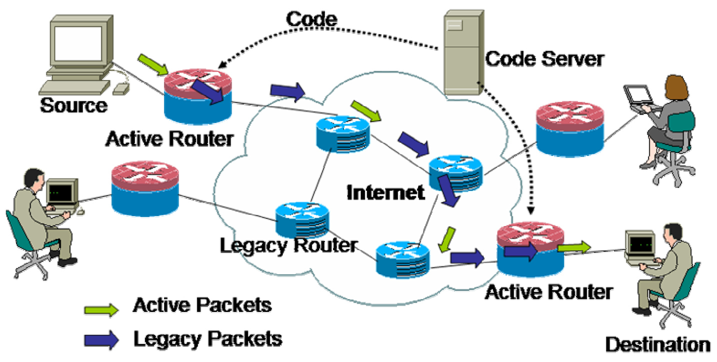


Fig. 1. Scenario of a multiservice network based on a programmable network

In some programmable networks, users can introduce their own executable codes into the active routers, but this is not a pragmatic solution because of the risk of introducing malicious codes. So, in order to allow the network administrator to control the codes running in active nodes, we propose to use code servers. Every active packet carries the code identifier, which identifies the code that the active routers must execute to process the active packet itself. When an active router receives an active packet, if it does not have the code to process it yet, it will download the code from a code server.

Active routers consume resources when the multiservice network offers the services demanded by the users. We will define a service model that forces the users to request a service before using it, so the multiservice network can accept or refuse the service according to the available resources. These services must be controlled in order to offer just the authorized services. So, active routers must only process the active packets that belong to an authorized service (e.g. transcodig service).

There are some approaches towards a programmable network in which active routers need to know who their closest active routers are in order to send them the active packets, while in other approaches active routers that process active packets do not need to know this information (its IP address). In this case active packets are sent to the destination and intercepted by the active routers in the path. We suppose that in a generic scenario of programmable networks, the active routers do not need to know the topology (the other active routers). This supposition allows us to propose a generic security solution valid for both programmable network technology approaches. In addition, it is a pragmatic requirement that the end systems do not need to know the active routers (its IP address) in order to send them active packets.

A programmable network could experience changes in topology which can be produced by changes in the network routes by new active routers that appear in the network or when an active router is down. The changes in topology can cause the changes to take place suddenly, as new active routers start to process the active packets of a service, or that other active routers suspend the processing of active packets. The security architecture must be immune to changes of topology.

It is assumed by the scientific community that the active routers will be located on the edge of the network, where a higher processing power to packet throughput ratio is possible. So, we consider that active routers will be located in the ISP networks that are on the edge of the Internet, which offer services directly to the users.

We propose to use SARA [4, 5] as active router because it follows some of the aforementioned ideas: (1) SARA uses a code server to download codes executed by the active routers, in this way codes can be controlled by network providers. (2) The SARA architecture allows upgrading legacy high-speed routers to work as active routers by delegating active processing to an external entity called assistant. (3) Active packets sent by SARA, have set the router alert [6] IP option. These active packets are sent between two end systems (Source and Destination), using the traditional IP routing. The router alert option allows the active routers in the path to catch active packets, process them and finally queue them in the router output to follow the journey towards their destination. This avoids costly tunneling management. (4) Active routers can be configured dynamically to pick up legacy IP packets compliant with a predefined pattern. The configuration of the pattern is carried out via active packets acting as control packets. The pattern can be specified using fields within the packet headers (e.g. source and/or destination IP address, transport protocol, source and/or destination ports…). The set of packets that match a specific pattern are called a flow.

3 Security Problem

Active routers must be protected against security attacks. An active router is more vulnerable than a legacy router because a active router processes external codes and active packets that may change its own behavior. The deployment of an active network must be carried out by authorized users, so active packets sent by these users, which allow the programmable network to be configured, must be authorized and authenticated. An efficient security mechanism of authorization and authentication has to be used so as not to consume too many resources of active routers. Code executed in active routers have to be downloaded securely from trusted code servers using services that guarantee authentication and confidentiality of codes.

In [3] we propose a security architecture to protect programmable networks such as SARA from malicious active packets and codes. This security architecture allows users to obtain authorization to send active packets from a specific source to a destination. So a user is authorized to obtain a certain service by sending active packets. Active packets and code are protected using cryptography.

Once active packets and code are protected, we focus attention on legacy packets, which are picked them up and processed by active routers. As we stated before, once a pattern has been configured, a predefined flow of legacy packets, which are sent from a source to a destination, are processed by the active routers in the path (e.g. by the transcoding service); we will call these legacy packets *legitimate packets*. A malicious entity could generate fake legacy packets that fulfill the specified pattern. By injecting these fake legacy packets into the same path as the legitimate ones (see fig. 1), the following active routers in the path will process them. So, these active routers use more resources than predicted. Hence, the injection of fake legacy packets can provoke a Denegation of Service (DoS) at active routers. We will call these fake legacy packets *inserted packets*.

The security goal here is to protect active routers. Cryptographic mechanisms could be used to identify the inserted packets, but we have rejected this option because the protection mechanism should be transparent to end systems applications. That is to say, legacy packets have to arrive to destination in a transparent way, even if the active router near the destination is down, the legacy-packet format must not be modified (e.g. introducing cryptographic information that the destination does not know how to process).

The state of the art in insertion attack detection usually uses mechanisms based on the observation of traffic at a point in the network, and the detection of an abnormal model of traffic behavior [7, 8, 9]. These techniques usually have a certain probability of making an error in the detection because they are based on a probabilistic interpretation of the observation. In this paper we propose a cooperative mechanism based on the observation of traffic at two points in the network. Furthermore, we use the programmable capability of the active routers to deploy this cooperative mechanism as a service. By using cooperative detection it is possible to obtain precise information in most cases, which allow us to reduce the risk of making an error in detection.

4 Protection Against Insertion of Packets

4.1 The Basic Description of Detection

The security mechanisms described in this section are for the detection of attacks based on the insertion of packets in a flow of legitimate packets that must be processed by active routers. This mechanism is applied to each segment of the network which is delimited by two active routers or by an end system with active application support and an active router. So, all the segments that make a path between the end systems can be protected.

At the starting point of each protected segment, a *Signalling Agent* (SA) counts the outgoing packets that belong to the observed flow. At the end of the segment, a *Monitoring Agent* (MA) counts the incoming packets that belong to the same flow. The counter of the packets is sent in a *Signalling Active Packet* (SAP) from the SA to the MA, intercalated among the legitimate flow's packets. So, the MA can verify whether the received counter is equal to the local counter at the moment of receiving the SAP.

A SAP is sent with a certain cadence of legitimate packets. The SAPs carry the current counter of outgoing legitimate packets at the SA and the cadence to send another SAP. The term cadence here refers to the number of packets that the SA has to count before generating the next SAP. If non-legitimate packets are sent during a period of time, a default SAP is sent with the counter at that moment, so a default SAP is sent at least at a predefined frequency. The counter or cadences that are carried by the SAPs may be modified by a malicious entity trying to avoid the detection of an attack. So, to prevent it, signalling packets must been protected against modification or fabrication. Because SAPs are active packets, protection of authentication and authorization mechanisms supported by the active network [3] are valid to protect signalling information.

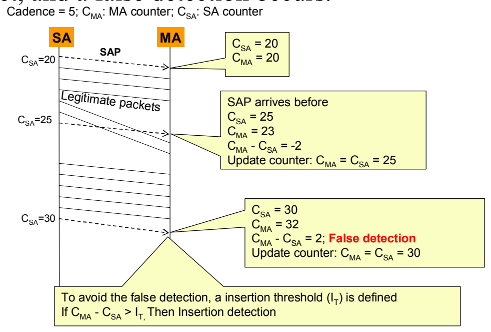
As an active router would be processing a lot of different flows of legitimate packets when an insertion attack is detected, the active router would prevent more packets from being processed than the previously predefined ones that belong to the attacked flow. So a procedure for discarding packets would be activated as a response against an insertion attack. The discarded packets must be selected from among the following packets that arrive at the active router and that belong to the attacked flow. In the discarding process, is not possible to identify the inserted packets, so any legacy packet (legitimate or inserted) could be selected to be discarded. When an attack is detected, because MA counter is bigger that SA counter, the MA counter is updated with the value of the SA counter.

4.2 Reordering and Duplication and Loss of Packets

Since within IP networks such as the Internet, packets can be reordered, duplicated or lost, we are now going to analyze the influence of these issues on the detection mechanism.

We can see the flow of legacy packets as a sequence of sets of legacy packets, where a set of legacy packets is separated by two consecutive SAPs. Because the network can reorder legitimate legacy packets or SAP packets, reordered packets would change to the previous or following set. As well as reordering, duplications of packets can be provoked because of malfunctions in nodes. This will provoke a false detection of an insertion in the MA.

Figure 2 shows that two legitimate packets of a set are delayed in the next set. In this case, the MA will detect that it lacks two packets when the second SAP arrives. Because the MA counter (C_{MA}) is updated with the value of the SA counter (C_{SA}), when the next incoming SAP arrives, the MA will detect that there are two extra packets in the following set, and a false detection occurs.



Cadence = 5; C_{MA}: MA counter; C_{SA}: SA counter
SA
MA
SAP
C_{SA}=20
Legitimate packets
C_{SA}=25
C_{SA}=30
C_{SA} = 20
C_{MA} = 20
SAP arrives before
C_{SA} = 25
C_{MA} = 23
C_{MA} - C_{SA} = -2
Update counter: C_{MA} = C_{SA} = 25
C_{SA} = 30
C_{MA} = 32
C_{MA} - C_{SA} = 2; False detection
Update counter: C_{MA} = C_{SA} = 30
To avoid the false detection, a insertion threshold (I_T) is defined
If C_{MA} - C_{SA} > I_T, Then Insertion detection

Fig. 2. Reordering of packets

Tests carried out in [10] show that reordering and duplication rarely happens. So the probability of obtaining a false detection resulting from these events is low. However, to reduce the risk of false detections, we define an *insertion threshold* (I_T) as the difference between the MA's counter (C_{MA}) and the SA's counter (C_{SA}). So, if C_{MA} - C_{SA} is higher than I_T, an insertion attack is detected. On the other hand, the insertion threshold would prevent legitimate packets from being discarded in the event of a small attack which does not mean a significant DoS problem for the active router.

When a packet is reordered, the number of packets skipped in the reordering process is usually small, 3 hops in 87% of reordering cases [11]. So, the minimum value of the insertion threshold should be small (3 or 4 are correct minimum values).

The loss of packets happens at a greater frequency than duplication and reordering. The loss of a legacy packet does not mean a DoS problem for the destination active router. The MA in this active router calculates the difference between the local and the received counters, and because $C_{MA} - C_{SA}$ is less than 0, no detection takes place. In this case C_{MA} is updated with the C_{SA} value. Even if an insertion attack compensates the quantity of packets lost, this does not mean a problem because the active router will not process more packets than the ones predicted.

A main problem happens when an SAP is lost. In this case, the MA does not receive the SAP at the expected time. The observation of this situation could be confused with a strong attack in which the MA receives a lot of inserted packets mixed with legitimate packets. In this case, the MA could assume that a strong attack may be taking place and lot of packets has been intercalated between two consecutive SAPs. These two situations are illustrated in figure 3a. The left-hand illustration shows that two consecutive SAPs have been lost and the MA has counted 11 packets from the last SAP received. The right-hand illustration in figure 3a represents a strong attack in which no SAP has been sent because the SA has only counted 2 outgoing legitimate packets (in both situations we use a cadence of 5). In this case the MA has counted 11 packets from the last SAP received, as there are 9 inserted packets. The MA cannot establish what has really happened until the next SAP arrives. So, in both situations the MA would assume the worst of the situations, which is a *strong insertion attack*.

We will consider that a strong insertion is taking place when a defined number (e.g. 3) of consecutive SAP packets do not arrive when they are expected. When strong insertion is detected, the response mechanism must be aggressive in order to save the active router integrity. An example of an aggressive response consists in discarding all packets that belong to the attacked flow until an SAP arrives and confirms that the attack has finished. But on the other hand, if false detection has taken place (because 3 consecutive SAP's were lost) an aggressive discarding takes place on the legitimate packets when no attack is really happening.

To reduce the impact of the false detection of a strong insertion, a *request for an SAP* packet (RSAP) is sent by the MA to the SA, when an SAP has not arrived as expected. The MA should receive an SAP every time a quantity of packets equal to cadence plus the insertion threshold has arrived (I_T), in the worst case. Every time this does not happen, an RSAP packet is sent to the SA.

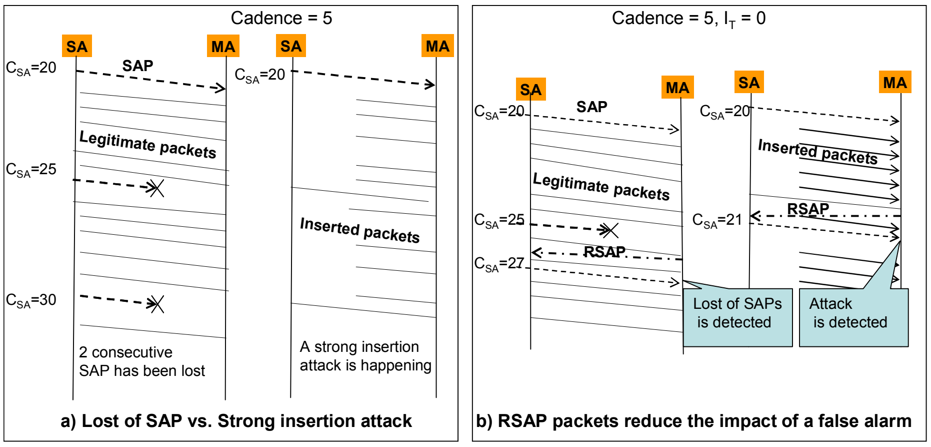


Fig. 3. Reaction against loss of packets

When the SA receives the RSAP, it sends a new SAP with the current value of the counter. So, when a strong attack happens, the SAP packets are sent to the MA more frequently and the attacks are detected beforehand. Figure 3.b illustrates this procedure in both situations, when a strong insertion attack happens and a false strong insertion attack is detected. For simplicity, in figure 3b we have supposed that cadence is 5 and the insertion threshold is 0.

We can see that when the MA has received 6 packets without an SAP, it sends the RSAP. In both cases (left and right), the SA sends an SAP in response to the RSAP received. When the new SAP arrives at the MA, it can be detected whether an attack is taking place or not. The RSAP is an active packet as is the SAP so it has the same authentication and authorization mechanisms that are necessary to avoid unauthorised use.

Summarizing the behaviour of the MA, two events can take place: 1) an SAP arrives and an insertion attack is detected because C_{MA}-C_{SA}>I_T or, 2) an SAP does not arrive when expected. In this case, the MA activates the state of strong insertion attack and sends an RSAP. If the state of strong insertion attack is active, when an SAP arrives at the MA, the following could happen: 1) if $C_{MA} - C_{SA} > I_T$ an insertion attack will be detected, then the MA will change to insertion attack state, or 2) if $C_{MA} - C_{SA} \leq I_T$ a loss of SAP is detected, then the MA will change to normal state.

4.3 Discarding Procedure

We have seen that the MA can produce two kinds of alarm: insertion attack and strong insertion attack. When an insertion attack happens, the MA knows how many packets have been inserted (C_{MA}-C_{SA}), so it can discard the same quantity of packets when the following packets arrive.

When a strong insertion has been detected, the mechanism used to discard legacy packets must protect the active router from DoS, being the least aggressive as possible with legitimate packets. Discarding all legacy packets until being sure that the attack has finished is good for the active router interest but it is an aggressive method on legitimate packets. We have defined a less aggressive *method based on memory*, which consists in discarding all legacy packets in the strong insertion attack state, but when an SAP arrives, it takes into account the amount of discarded legacy packets and discounts it from the amount of legacy packets to be discarded (defined by C_{MA}-C_{SA}). This method is fairer for legitimate packets but it has problems because it produces strong peak in traffic passing through the router (see figure 4).

To find a solution to the peaks appearing in the method based on memory, we propose a discarding *method based on a filter* that consists of measuring the traffic rate when no attack happens. When a strong attack takes place, the MA discards all the packets to maintain the last measured traffic rate. We use a discrete low-pass filter to obtain an estimation of the traffic rate. Some tests have been carried out using different filter coefficients to obtain a good behaviour for discarding when a strong attack takes place. The advantage of discrete filters is that they are easy to implement and the processing cost is low compared to the rest of MA process.

5. Validation Tests

Different tests have been carried out to validate the proposal. At first, the proposed mechanisms were simulated using the simulation tool ns-2 (*Network Simulator v2*), in order to analyze their behaviour in different situations, and then validate them. Then, an implementation was carried out in order to measure the resource consumption and to test its behaviour in a real scenario by tuning parameters as coefficients used for the discharging filter.

The simulation scenario consists of two active routers and a malicious node located between both active routes in order to simulate the loss of packets and insertion attacks. A lot of simulations have been carried out by mixing different situations: different traffic models, different loss of packet levels, and different attack intensities. The results obtained demonstrate that the mechanism works well in different situations and provokes quite a few false detections, even if the percentage of lost packets is high (2 false detections in 10,000 seconds of observation with a loss of 20%). So the mechanism is resistant to the loss of packets.

The second set of tests tries to verify whether the consumption of resources is proper and scalable. Mechanisms have been implemented for this proposal. The required time for MA and SA to process both the legitimate and SAP packets is 20% less than well-known and the most efficient cryptographic mechanism based on processing packets using HMAC-MD5. Furthermore, we have verified that the processing time increases linearly with the increase in the attack intensity, so the system scales properly.

We have compared the discarding method behaviour using a filter with the discarding method behaviour based on memory. The tests have been carried out on a real Internet scenario using two end points connected to the Internet. The legitimate packets follow a Pareto distribution using an ON/OFF period of 35/15ms and rate of 64 Kbps. We have provoked attacks using CBR traffic of 64 Kbps for 600 seconds. The percentage of loss of packets that the network causes during the test was 6.8%.

In both discarding methods, figure 4 shows the outgoing *non-discarded traffic* compared to the *legitimate traffic* coming into the MA, when an attack is taking place.

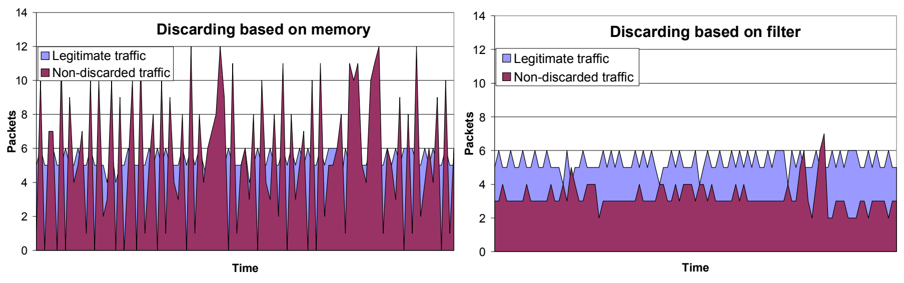


Fig. 4. Outgoing traffic compared to legitimate incoming traffic at the MA

We can see that the non-discarded traffic in the case of discarding based on memory is less stable than the case of discarding based on filters, because a multiple peak of traffic appears that overtakes the incoming legitimate traffic. So, the method based on filters is more conservative to the active router interest, which is the main concern of the security mechanisms. Furthermore, we have seen that discarding based on filters is a little less aggressive to the legitimate traffic than the discarding method based on memory. Finally, the processing time when a packet arrives at the MA increases by 11.8% using filters rather than the memory-based method. We think that is a reasonable cost.

6. Conclusion and Future Work

A model of a multiservice network based on programmable networks has been proposed in this paper. The active routers must be protected against a DoS attack that consists of the insertion of false legacy packets. We have proposed new mechanisms to tackle this problem based on the cooperation between active routers and using the capability of exchanging secure signaling information between active routers. We have carried out tests to validate the mechanisms, to verify the proper consumption of resources, and their scalability. We have proposed mechanisms to discard the packets in case of a strong attack. Finally we can conclude that the proposed mechanism against insertion attacks has a reasonable consumption of resources, and is pragmatic enough to be applied to the scenario described.

For future work we are interested in identifying the legitimate packets in order to carry out a selective discarding. So we are working on marking the outgoing legitimate packets at the SA, using hidden information in active packets to synchronize the SA and the MA.

References

[1] D. Wetherall, U. Legedza, J. Guttag, *Introducing New Internet Services: Why and How,* IEEE Network, Special Issue on Active and Programmable Networks, vol 12, no 3, May/June 1998, pp 12 -19.

[2] D. Larrabeiti, M. Calderón, A. Azcorra, M. Urueña, *A practical approach to Network-based processing*. IEEE 4th International Workshop on Active Middleware Services. 23 de Julio, 2002. Edinburgo, Escocia. TIC2001-1650-C02-01

[3] B. Alarcos, M. Sedano, and M. Calderon, *Multidomain Network Based on Programmable Networks: Security Architecture.* ETRI Journal, vol 27, no 6, pp. 651-665. Dec. 2005.

[4] GCAP: Global Communication Architecture and Protocols for new QoS services over IPv6, IST 1999-10504-GCAP project. http://www.laas.fr/GCAP/

[5] A. Azcorra, M. Calderón, D. Larrabeiti, M. Urueña, *Software Tools for Networking: Simple Active Router Assistant (SARA)*, IEEE Network, Vol 16, N0 4, July 2002.

[6] D. Katz, *IP Router Alert Option*, RFC2113, February 1997.

[7] Thomer M. Gill and Massimiliano Poletto, *MULTOPS: a data-structure for bandwidth attack detection.* In proceedings of the 10th USENIX Security Symposium, August 2001.

[8] Jelena Mirkovic, Gregory Prier, Peter Reiher, *Source-End DDoS Defense.* Second IEEE International Symposium on Network Computing and Applications, April 2003.

[9] Seong Soo Kim, A. L. Narasimha Reddy and Marina Vannucci, *Detecting Traffic Anomalies at the Source through Aggregate Analysis of Packet Header Data*. May 2003. http://dropzone.tamu.edu/techpubs/2003/TAMU-ECE-2003-03.pdf.

[10] Sharad Jaiswal, Gianluca Iannacconne, Chirstophe Diot, Jim Kurose, *Don Towsley, Measurement and Classification of Out-of-Sequence Packets in a Tier-1 IP Backbone*. In proceedings of the IMW 2002, ACM Press, November 2002.

[11] John C. R. Bennet, Craig Partridge and Nicholas Shectman, *Packet Reordering in not Pathological Network Behavior*, IEEE/ACM Transactions on Networking, Vol. 7, Nº 6, pp. 789-798, December 1999.