Analysis of Denial of Service Attacks on Mobile Ad-Hoc Networks

Heshem A. El Zouka

***Abstract*—The mobile ad hoc networks are highly vulnerable to attacks because of its unique characteristics such as: open network architecture, shared wireless medium, and stringent resource constraints. These attacks can potentially degrade the network performance by constantly transmitting packets and keeping the ad hoc network busy. This paper presents the review and the comparison between existing variants of TCP protocols, such as TCP Tahoe, TCP Reno, New Reno, SACK TCP and TCP Vegas with regard to their security implications. A properly-designed and implemented security measures can significantly reduce the load that the malware places on the mobile Ad-Hoc Networks.**

***Index Terms*—Ad hoc networks, congestion control, security protocols, TCP protocols.**

I. Introduction

The mobile ad hoc networks are formed dynamically by mobile wireless hosts. Thus, there would not be any need for a central controller or standard support devices which are available in a traditional network, and therefore forming an infrastructure-less wireless network which builds, operates and maintains with the help of intermediate wireless nodes. Due to their limited transmission range, these mobile nodes depend on surrounding nodes to forward packets and maintain routes [1].

The security services must be distributed, cooperative and consistent with the available bandwidth. The traffic of the attack uses the bandwidth to process resources at the target node and the network itself, so that legitimate user will be unable to access the ad hoc network.

The bandwidth consumption attack floods the entire network with necessary traffic that will definitely prevent legitimate user from reaching specific network resources or services. A resource consumption attack (RCA) is another attack which ties up the resources of a victim system and targets a server or a process in the victim network and makes it unable to exchange request-routing information.

Any amount of resources can be flooded with a considerably strong attack and solving this problem requires a defense mechanism that will detect the attack and respond to it by dropping the excess traffic as soon as it is detected.

The rest of the paper is organized as follows: the second and third Sections describe the TCP performance of different TCP protocols and their security analysis respectively. A proposed defense algorithm is presented with the simulation results and discussion. Finally, Section VI presents the conclusions.

Manuscript received July 1, 2015; revised December 1, 2015.

Heshem A. El Zouka is with the Department of Computer Engineering, College of Engineering and Technology, Arab Academy for Science and Technology and Maritime Transport, Alexandria, Egypt (e-mail: helzouka@aast.edu).

II. Performance Analysis of Mobile Ad-Hoc Networks

The Transmission Control Protocol (TCP) provides a reliable end-to-end transport service in high speed networks. TCP performance should be optimized, as it is the main factor in the process of Internet traffic routing. This performance is mainly enhanced by the performance of the congestion control algorithm it employs. This section of the paper presents the review and comparison between existing variants of TCP protocols, such as TCP Tahoe, TCP Reno, New Reno, SACK TCP and Vegas [2]. The characteristics of TCP vary in terms of the type of TCP variant used. For example, TCP Tahoe is based on the conservation of packets principle; if this principle were obeyed congestion avoidance would then become the issue, because the connection is used for the entire conversation as indicated in the following pseudo code:

```
CWND= MSS
if  tcp.acks = = dup_rcv.acks
  dupacks++;
     if dup_rcv.acks = = 3
        retransmitsegment(snd.acks)
        ssthresh = max(CWND/2 , 2*MSS)
CWND= ssthresh;
    else
       dupacks=0:
```

However, if the connection runs at the available bandwidth capacity, then a packet will not be injected to the network unless guaranteed reserved packet istaken out as well. This principle is carried out when TCP uses the acknowledgement mechanism to check the arrival of data and to confirm the receipt of those packets that have reached the destination within fixed time. For each connection, the TCP acknowledgement maintains the congestion window and limiting the total number of unacknowledged packet to utilize the network capacity efficiently [3]. However, TCP Tahoe has a few disadvantages. First of all, it takes a complete RTT to detect each packet loss and it takes even longer time in most implementations to recover packet loss.

Secondly, it sends cumulative acknowledgments which provide little information and consequently follows a "go back N" or a retransmission approach. Therefore, every time a packet is lost it waits for a RTO time which offers a major cost in high bandwidth delay product networks.

The performance of Vegas protocol is also compared with

DOI: 10.7763/IJET.2017.V9.937

New-Reno. Besides the ability of Vegas to prevent coarse-grained timeouts of New-Reno, it does not need to wait for three duplicate acknowledgments before it retransmits a lost segment. Its congestion avoidance algorithm is very efficient when utilizing ad hoc network resources and detecting any congestion.

Alternatively, TCP Vegas uses the congestion avoidance algorithm to prevent packet loss by decreasing its (WND) as soon as it detects an incipient congestion Thus, the size of the congestion window is determined by the difference between actual throughput and estimated segment throughput as follows:-

Expected Rate (e) = CWND / BaseRTT
Actual Rate (a) = CWND / RTT
if (e = = a) || (a<a)
cwnd = BWE x RTT ; BWE (estimated BW)
else
CWND = β x (BWE x RTT) ; 0< β < 1

TCP Vegas defines two thresholds denoted as α and β. Both of these values are used for controlling the congestion window, which is changed as follows:

$$\text{CWND} = \begin{cases} \text{CWND} + 1 & \text{diff} < \propto \\ \text{CWND} & \propto \leq \text{diff} \leq \beta \\ \text{CWND} - 1 & \text{diff} < \beta \end{cases}$$

If Diff is < α, TCP, Vegas estimates the absence of congestion and increases cwnd by 1 in the next round. However, if Diff > β, TCP Vegas expects an impending congestion and decreases cwnd linearly in the next round. Else, it leaves the congestion window unchanged if there are enough buffer in the intermediate routers. If there is enough buffer space, the Vegas congestion avoidance algorithm can function effectively with a high throughput rate and a fast response time.

Clearly, RCP Vegas congestion avoidance mechanism is less effective when the load increases or the number of router buffer decreases.

Also, New-Reno sender has to wait for one RTT to detect each packet loss. When the acknowledgment for the first retransmitted segment is received, only then the New Reno can come out of fast recovery phase and deduce which other segment was lost [4]. This obviously leads to redundant retransmission and degrade the network since losses between 10% and 20% of the total transmitted packets will affect the performance of the ad hoc network significantly.

Obviously, Vegas modifies New-Reno in the sense that the packets can be retransmitted with fewer than three duplicate ACKs, which may significantly reduce the probability of getting a run-time-out by 15% (instead of 25% in Reno).This makes TCP Vegas achieves better throughput than the standard TCP Reno, with less packet loss and hence better utilizing the bandwidth potential of the links [5], [6].

TCP SACK is an extension of the TCP New Reno. It only modifies the fast recovery algorithm of New Reno while keeping other algorithms unchanged. Similar to TCP Reno, SACK deals with multiple packet losses from the same window but it has the benefit of using selective acknowledgement of packets instead of using cumulative acknowledgement mechanism as contained in Reno and Tahoe.

However, the biggest drawback of using TCP SACK is the difficulty of implementing the selective acknowledgement, since a set of additional control fields are needed to acknowledge the selective segments at the receiver and a sender side, which is not an easy task, as it has to modify the TCP protocol implementation for all mobile nodes. Moreover, TCP Vegas is much better than TCP SACK, as it provides efficient estimation of incipient congestion by the means of measuring any change in the throughput rate.

III. Security Aspects of Mobile Ad-Hoc Networks

Wireless networks are frequently exposed to many security problems, because the intrusion on the transmission medium is easier than that on wired networks and it is possible to conduct denial of service attack (DoS) by simply scrambling the used frequency bands.

The DoS attack is considered one of the most serious attacks in mobile ad hoc networks and most proposed protocols to defend against this attack have met with failure due to node movement, lack of wireless connection, and scalability issues. In addition, the attacker can easily attack one single physical device in the mobile ad hoc network in order to launch a coordinated attack on the whole available resources of the network. This serious attack starts when a large volume of segments is sent to a victim machine through the simultaneous cooperation of a large number of nodes that are distributed through the network.

In general, DoS attack techniques can be grouped into three main scenarios. The first attack scenario targets storage and processing resources in ad hoc nodes; it targets the memory, storage resources, and aggregates the computing power of the mobile devices. In other words, the malicious node continuously sends a stream of flooding packet to its surrounding nodes in attempt to overload the storage space and exhaust the memory space of these nodes. Obviously, this will prevent the legitimate nodes from transmitting or accessing the network services.

The second attack scenario targets energy resources, specifically the battery power of the ad hoc nodes. A malicious node can perform a TCP flooding attack by consuming the victim's battery energy and prevent other nodes from communicating with the legitimate nodes. Network monitoring tools will be needed in detecting such malicious nodes and preventing their consequences.

Finally, the third attack scenario targets the network bandwidth or connectivity. Bandwidth DoS attack overflow the ad hoc network with a high volume of traffic using existing network resources causing legitimate nodes of the network to be unable to communicate. Connectivity attacks overflow a node with a high volume of connection requests consuming all available network resources, so that the node cannot process other legitimate node requests. If an attacker is located between two wirelessly communicating nodes, impersonating both legitimate nodes, the attacker can control these nods and the high speed communication link between them in such a way that the attacker can waste the

network bandwidth and disrupt the service for other nodes. This will overload the network traffic causing a significant performance degradation.

IV. Reliability of Mobile Ad Hoc Networks

As Vegas is much more powerful in facing and detecting lost segments, it also has experienced fewer retransmissions in the event of connection in order to avoid unnecessary congestion control invocation. Vegas is also has better performance on congestion avoidance and therefore uses network resources more efficiently. However, one drawback of this protocol is its weakness against Pulsing Denial of Service (PDoS) attack. In this attack, the attacker sends undetected sequence of pulses in attempt to reduce the TCP throughput. [7] , [8].However, in comparison to TCP Reno, Vegas can prevent more than half of the course- grained timeouts of Reno, as it detects and retransmits more than one lost packet before timeout. Vegas does not need to wait for three duplicate packets in order to transmit faster. It has also a security model that leads towards sudden changes in traffic pattern [9]. Also, TCP Reno uses this detection method to identify an unclassified threat. In some other work, such as [10] and [11], they proposed a new secure-power-aware ant- routing algorithm that is inspired from ant colony optimization and known as SPA-ARA algorithm. TCP RENO uses the packet losses as an indication for network congestion and performs very well when the packet losses are too small. But if there is multiple packet losses, then RENO does not perform well and its behavior is exactly the same as Tahoe. Another problem of Reno is that if the window is very small, then it would never receive enough duplicate acknowledgements to trigger fast retransmit. Hence it has to wait for a time out to retransmit the lost packet. Thus it cannot eliminate packet losses effectively.

In order to eliminate DoS attack, authors have proposed a DoS mitigation technique that uses digital signatures to verify legitimate packets and drop all the packets that fail the verification [12]. Some other studies observed a new distributed denial of service attack in mobile ad hoc networks which is known as a folding based DoS attack [13]. The new attack can result in denial of service if it is used against both on-demand distance vector (AODV) and dynamic source routing (DSR) protocols. Other researchers have proposed statistical analysis approach to detect routing attack by providing early detection of flooding DoS attacks [14]. Also, [15] has proposed an intrusion detection/response framework for mobile ad hoc networks.

All mobile nodes in any ad hoc network will carry a client process or a server process as some services require high power levels. Thus, node may communicate within a limited range and it may fail temporarily. Transmission distance may go up to 100 meters (line-of-sight RF) for outdoor applications relying basically on the power output and environmental characteristics. On the other hand, the range often goes down to less than to 10 meters for indoor applications as it relies on the number of walls in between the two communicating entities. Then, power consumption should be reduced by operating servers if they were needed. Each node, then, will try to reach global objectives like maintaining communication connectivity. The server will face two problems, then, even if the current network topology and the availability of the nodes were given. First, which node should act as a server, based on processing load, bandwidth availability, connectivity, and battery power of the node, hence all these factor control the choice of the node to be act as a server node. Secondly, deciding which servers are required to meet current demands of client nodes. Clearly, mobile ad hoc networks are much more exposed to attacks than conventional wired networks due to their dynamic nature, and any node can join or leave the network without permission. Thus, the security issues are considered the main challenge facing ad hoc networks.

V. Defense Algorithm and Simulation Results

A new defense mechanism is here. It involves a little change of implementation in the victim's server and it does not need the TCP packet format nor the router to be modified. Moreover, the complexity of the algorithm is in the order of packet transmission time and no segment of the previous connection which has been destructed will be restored.

The segment can only be received after the victim's server crashes and then recovers all these TCP connections within maximum segment life time [16], which is impractical. Hence, all other flows will considered as false positive and within that algorithm.

The victim's server generates tow random numbers r1 and *r*2 automatically in order to manipulate the number of packets send and received in a way that protects the victim's server from DoS attacks in general and optimistic acknowledgment attacks in particular. These two numbers are randomly selected from n integer and the range of each number is chosen to be in the range of 1 and 100 only. As for he function rand( ), it initializes the random number generator which applies a time function to generate these number of bytes specified in the TCP Maximum Segment Size (MSS). The victim's server will check the acknowledgment number to see which packets are actually acknowledged if a DoS attack ever take place.

The attacker won't know which packets have been sent and how much less data the server will send as soon as the MSS message is sent. It will think that each segment has a fixed number of bytes and, then, create optimistic acknowledgments which could be easily detected on the server side. If the MSS is a multiple of multiple of 1024 bytes, and the default MSS advertised is 4096 bytes, for example, the MSS is sent with a different size based on the following formula: MSS (r1) = MSS – int (r2/2), where r1 represents the MSS sequence number and r2 represents the number of bytes reduced each time a new segment is sent. According to the receiver, the attacker replies acknowledgment prior to actually receiving the MSS it acknowledges. If the victim's server sends data [1024:2048], the attacker may immediately ACK 2560 without actually receiving data [2048:2560] to induce DoS attack by increasing the CWND arbitrarily rate. Consequently, the proposed algorithm would detect optimistic acknowledgement attacks efficiently and resist DoS attacks, and the detection rate of the proposed algorithm is relatively

high among distributed mobile ad hoc networks. According to the receiver, then, the attacker needs to guess two numbers correctly in order to escape the detection. The probability of guessing two numbers with each number randomly chosen from 1 to 100 will apparently be "1 in a million" chance before the attacker can succeed in comprising the system.

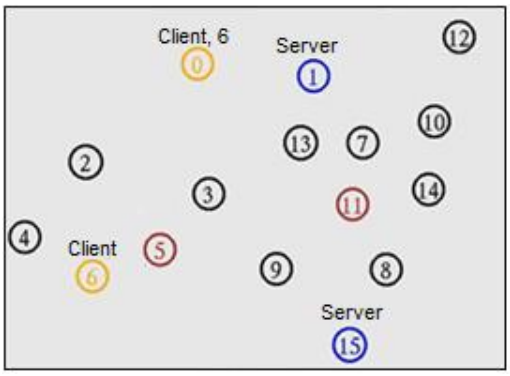

Fig. 1. Topology of ad hoc network inNS2 environment.

The proposed algorithm was implemented using NS2 simulator environment [17], and its performance against DoS attacks was evaluated. First, 15 nodes were randomly deployed over an area of 100x100 square meters. Then, the simulation was run for 50 sec and average of the results was taken. The client nodes are respectively represented by 0, 5, 6 and 11 nodes while the servers are represented by nodes 1 and 15, respectively as shown in Figure 1. Data packets and their ACKs are assumed to take different path from source to destination without any segment loss or errors in case of protected channel. All of these results are presented in large variations of RTT values measured under different traffic scenario. The analysis results guide us to conclude that the average packet loss is less in case of TCP Vegas compared with other TCP variants. However, when increasing the number of nodes, the signal to noise ratio increases in case of all TCP protocols, but decreases in case of TCP Reno. Obviously, the numbers of packet losses vary with the increase number of ad hoc connections. In addition, the simulated routers might fail to deliver (drop) some packets in case of DoS attack. Therefore, some packets might be dropped, depending on the traffic delivery state of the network as shown in Fig. 2.

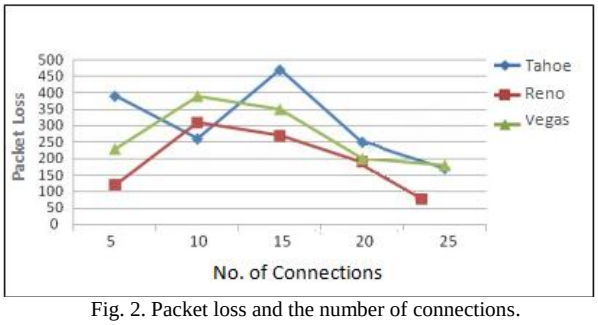

Fig. 2. Packet loss and the number of connections.

The TCP Protocol may ask for retransmitting these delayed data, which cause to minimize the transmission rate overall the whole network. The number of the dropped packets, which caused by DoS attack, can be computed by the following formula:

no of dropped pkets = no of the snd_pkets – no of rec_pkets

In Fig. 3 and Fig. 4, the average throughput and packet delivery ratio were plotted respectively. As noted earlier in this paper, large variations are observed in the graph because the added level of security may affect the performance of TCP in ad hoc networks.

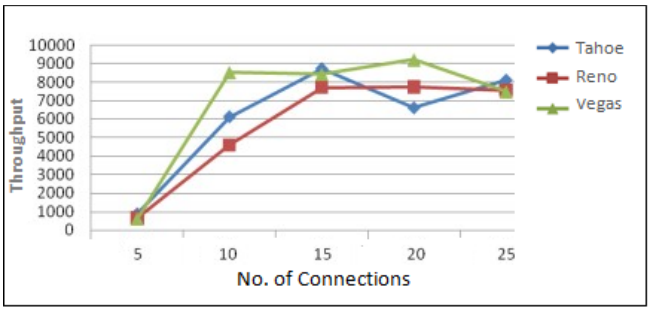

Fig. 3. Throughput under various number of connections.

The following simulation parameters were used and the ad hoc network performance for three different scenarios was studied. The first scenario involves a large number of connections between source and destination that have been built on the fly and without the presence of attack.

The second scenario involves a large number of connections, where the attack is conducted and still in progress. The third scenario shows the throughput status when there was no DoS attack on the ad hoc network globally going on. It has been observed that the first scenario does not incur any packet loss since it maintains connectivity between user nod, and its performance is very similar to result of the third scenario which does not suffer from DoS attack.

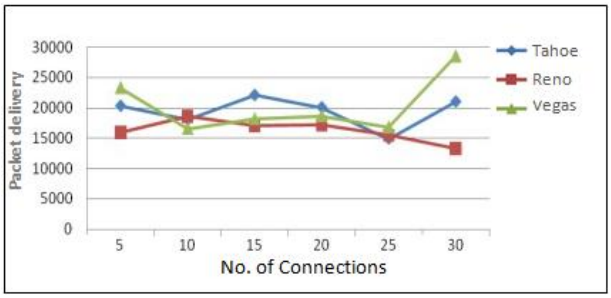

Fig. 4. Packet delivery under various number of connections.

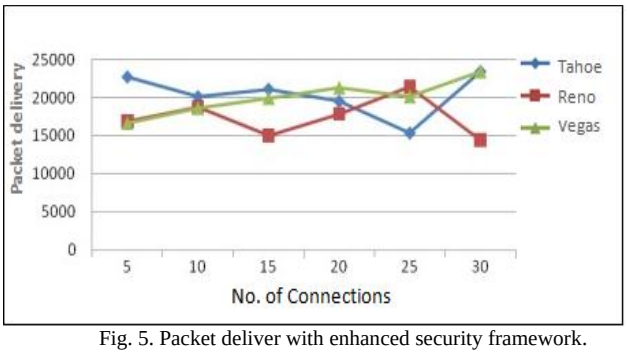

Fig. 5. Packet deliver with enhanced security framework.

The throughput achieved for different implementations of TCP protocols along with their security framework are also shown in Fig. 5. As indicated in the figure, the number of packet losses varies with the total number of mobile nodes involved in communication. TCP Tahoe produces and

average total loss rate of 12.75 % in the transmitted packets, while TCP Reno generates 17.28%.

However, TCP Vegas provides less packet loss and better overall throughput than other TCP variants. With regard to throughput rate, TCP Vegas provides better simulation results compared with other TCP variants. This is because the recover mechanism used by TCP Vegas allows it to detect network congestion earlier than TCP Tahoe and TCP Reno, although the latter two perform better with short packets.

VI. Conclusion and Future Work

In this paper a review of transport layer protocols developed for ad hoc networks was presented. First, the performance of various TCP protocols was compared with respect to congestion control, reliability, and security. Then, the suggested solutions for lack of bandwidth and the challenges in improving TCP performance by employing different strategies over a number of existing TCP variants used in mobile ad hoc network was surveyed. Then, the impact of DoS attack was analyzed and a defense mechanism that improves the performance of the network was proposed. The proposed algorithm using NS2 simulator environment was implemented, and it was observed that the number of packet losses varies with the total number of mobile nodes involved in communication. It was also found that TCP Vegas provided less packet loss and better overall throughput than other TCP variants.

References

[1] M. K. J. Kumar and R. S. Rajesh, "Performance analysis of MANET routing protocols in different mobility models," *International Journal of Computer Science and Network Security*, vol. 9, no. 2, pp. 22-29, Feb 2009

[2] S. Floyd and K. Fall, "Simulation based comparisons of Tahoe, Reno, and SACK TCP," *ACM Computer Communication Review*, vol. 26, no.3, pp. 5–21, July 1996

[3] P. Tomar and S. Yadav, "Enhanced reliable TCP for congestion control with corruption control in MANETs," *Journal of Global Research in Computer Science*, vol. 3, no. 4, April 2012.

[4] N. Parvez, A. Mahanti, and C. Williamson, "An analytic throughput model for TCP new Reno," *IEEE/ACM Trans. Networking*, vol. 18, no. 2, pp. 448-461, April 2010.

[5] H. Lee, S. Lee, and Y. Choi, "The influence of the large bandwidth-delay product on TCP Reno, new Reno, and SACK," in *Proc. Information Networking Conference*, Oita, Japan, Feb. 2001, pp. 327–334.

[6] B. Qureshi, M. Othman, and N. Hamid, "Progress in various TCP variants," in *Proc. 2nd International Conference on Computer, Control and Communication*, 2009.

[7] J. Andrews *et al*., "Rethinking information theory for mobile ad hoc networks," *IEEE Communications Magazine*, December 2008.

[8] A. Afanasyev, N. Tilley, P. Reither, and L. Kleinrock, "Host-to-host congestion control for TCP," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 3, Aug 2010

[9] L. S. Brakmo and L. L. Peterson, "TCP Vegas: End to end congestion avoidance on a global internet," *IEEE Journal on Selected Areas in Communication*, vol. 13, pp. 1465-1490, 1995.

[10] S. Agrawal and S. Singh, "An experimental study of TCP's energy consumption over a wireless link," Vienna, Austria, Feb. 2001.

[11] H. Singh and S. Singh, "Energy consumption of TCP Reno, new Reno, and SACK in multi-hop wireless networks," June 2002.

[12] S. Floyd and T. Henderson "The New-Reno modification to TCP's fast recovery algorithm," *RFC 2582*, Apr. 1999.

[13] D. Bisen and S. Sharma, "Improve performance of tcp new reno over mobile ad hoc network using abra," *International Journal of Wireless and Mobile Networks*, vol. 3, no. 2, April 2011.

[14] H. Singh, S. Saxena, and S. Singh, "Energy consumption of TCP in ad hoc networks," vol. 10, issue 5, Sep. 2004.

[15] A. Gurtov, "Effect of delays on TCP performance," in *Proc. IFIP Personal Wireless Communications*, 2001.

[16] V. Tsaoussidis *et al*., "Energy/throughput tradeoffs of TCP error Control strategies," France, July 2000.

[17] (2014). Network simulator. [Online]. Available: http://mohit.ueuo.com/NS-2.html

Heshem A. El Zouka is an associate professor at the Arab Academy for Science and Technology, the Department of Computer Engineering, College of Engineering and Technology, Alexandria, Egypt.

He received his PhD in computer networks and security from Nottingham University from 1999 to 2006. His research interests include algorithms, signal processing, simulations, programming, machine learning.