Developing Attack Defense Ideas for Ad Hoc Wireless Networks

R. Oliveira[1,2], B. Bhargava[1], M. Azarmi[1], Ed' Wilson T. Ferreira[2], W. Wang[3], M. Lindermann[4]

[1]Department of Computer Science
Purdue University
West Lafayette, USA
e-mail: ruy,bb,mazarmi@cs.purdue.edu

[2]Department of Computer Science
Federal Institute of Mato Grosso
Cuiabá, Brazil
e-mail: ruy,ed@cba.ifmt.edu.br

[3]Department of Software and Information Systems
University of North Caroline at Charlotte
Charlotte, USA
e-mail: weichaowang@uncc.edu

[4]Air Force Research Laboratories
Rome, NY, USA
e-mail: lindermanm@rl.af.mil

***Abstract*—Ad hoc networks are natively cooperative systems in the sense that their nodes have to relay data to one another. The inherent drawback of this scheme is that it renders these networks susceptible to intruders. Collaborative attacks, in which various attackers may coordinate actions to hit the network stronger, are also facilitated by the natural cooperation existing in ad networks. In this paper, we discuss the most important forms of attacks, address possible collaborations among attackers, show how machine learning techniques and signal processing techniques can be used to detect and defend against collaborative attacks in such environments, and discuss implementation issues. We also perform evaluations to determine the best design options for our preliminary proposed scheme to collaboratively respond to attacks.**

Keywords- Collaborative attacks, Machine learning, Signal processing, Collaborative defense.

I. INTRODUCTION

Over the last few years various research work have been conducted toward securing ad hoc wireless networks. Most of such efforts have been put on mechanisms to detect attacks in these networks [1],[2],[3]. Not much has been done in terms of response mechanisms for defending ad hoc networks, though. In [4] we discussed the main forms of attacks ad hoc networks are vulnerable to. We also emphasized the importance of collaborative defense strategies. As intruders improve their ability to collaborate toward more devastating attacks, the intrusion response systems should also react in a coordinated manner. This would render the collaborative defending system faster.

Before the Intrusion Response System (IRS) takes action, the Intrusion Detection System (IDS) has to infer an underway attack. In this sense, the IRS performance depends strongly on the detection time of the IDS in place. Although both mechanisms, IDS and IRS, are normally treated as a single security system, for the sake of clarity we address them separately in some parts of this work.

Various approaches for IDSs have been proposed in recent years. In general, the IDSs are classified into two categories: anomaly detection and misused-based detection. The former corresponds to the IDSs that keep track of the regular behavior within the network and imply attacks whenever significant deviations from the regular behavior happen. The latter is associated to the IDSs that keep comparing the actions inside the network to previously known attacks patterns. The main problems with these two approaches are: the anomaly detection approach may render high number of false positive, as deviations from regular behavior are not always really linked to attacks; the misuse-based detection approach needs a database to store the known attack patterns (signatures), which has to be updated continuously. Hence, a tradeoff between these two approaches is needed.

An interesting strategy to reduce the number of false positive in the anomaly detection approach is to deploy collaborative IDSs. For instance, each IDS may send its triggered alerts to a central module which correlates the incoming alerts of all IDSs and generates a more elaborated and general alarm to the whole system [5]. Our work makes use of this technique, as shown later.

Collaborative IDSs can also play a key role in speeding up intrusion detection in the misused-based detection approach. If the IDS in a given node detects the intrusion and shares the information with the IDSs in other nodes, then not every IDSs will need to perform the pattern matching which is both complex and time-consuming [6].

More general approaches including both detection techniques above are interesting as well. For example, in [7] a collaborative intrusion detection system (CIDS) allows for different sorts of IDSs to work cooperatively. It uses a central module that receives the alerts of the IDSs spread across the network and combines them toward a robust decision about the intrusion. The decision is taken based on graph-based and Bayesian concepts.

Concerning the Intrusion Response System - IRS, most existing approaches are not automated, i.e., the system administrator is normally in charge of manually triggering the response to the attack [8]. Several factors have to be taken in account in designing an efficient IRS. For instance; proper decision criteria for blocking an intruder or isolating contaminated hosts have to chosen; the effectiveness of collaboration among distributed IRSs has

to be assured; the severity degree of the attack underway may be used as a metric; the IRS may deny access to legitimate nodes if inappropriate metrics threshold are used to trigger a response; etc.

To the best of our knowledge, a comprehensive study on IRS in ad hoc networks is still missing. We proposed in this paper to address collaborative IRS in these networks. The idea is to discuss the main involved issues, to propose conceptual schemes for improvements, and outline guidelines for future work.

II. Collaborative Attacks in Ad Hoc Networks

The most well-known attacks in ad hoc networks include: (1) *blackhole attack* [9] in which a node transmits a malicious broadcast informing that it has the shortest path to the destination aiming to intercept messages; (2) *wormhole attack* [10],[11] where an attacker records packets (or bits) at one location in the network, tunnels them to another location, and retransmits them there into the network; (3) *DoM attacks* [12] where malicious nodes may prevent some honest ones from receiving broadcast messages by interfering with their radio; and (4) *sybil attack* [13] where a malicious user obtains multiple fake identities and pretends to be multiple, distinct nodes in the system., so it can control the decisions of the system, especially if the decision process involves collaboration for voting; rushing attack [14] where malicious nodes forward the RREQ packets, asking for a route, to the destination node quicker than the legitimate nodes do. This is possible because the legitimate nodes only forward the first received RREQ packet for a given route discovery. Besides, the attackers can tamper with either the MAC or routing protocols to get faster processing. As a result, the path through the malicious nodes is chosen, which can cause large throughput degradation.

We understand that if some intruders collaborate, the resulting aggregate attack can become either stronger or weaker. This will depend on the needs of each kind of attack. Examples of incompatible forms of attacks are the wormhole and DoM attacks. While the former typically needs a fast connection, to be attractive to the routing algorithm in place, the latter reduces bandwidth. Thus, the defense mechanism does not need to have an explicit algorithm to deal with this unlikely combination. As a result, there is a benefit in terms of energy consumption, which is vital for the normally battery-powered devices in place.

On the other hand, there are some combinations of attacks that might become very successful. As an example, Fig. 1 [4] illustrates a situation in which two attacks take place simultaneously. Node A perpetrates a blackhole attack and nodes X and Y collude to carry out a wormhole attack. In this scenario, if node A and X collaborate, then the data packets from node S will be forwarded through the tunnel, as shown in the Figure. Node A will receive a route request packet (RREQ) from node S and will reply with a route reply packet (RREP) stating maliciously that it has the shortest path to node D. Then node A will establish a route through node X which will build a tunnel to node Y, so the communication between the two end nodes (S e D) will be established through the path, including the tunnel, as depicted in Fig. 1.

With this setup, as node A does not drop packets, it will go undetected by various existing proposals for blackhole attacks. Nodes X and Y will receive every packet of the connection and can tamper with their contents or simply selectively drop them [11]. In this case, in order to be the selected path by the routing protocol, the tunnel does not need to be really attractive to the routing protocol, as assumed by most related work. This is facilitated here by the malicious node A. This particular example shows that such a combined attack has to be addressed particularly. The defense mechanism has to be smart enough to take tailored actions for each threat announced by the classification mechanism.

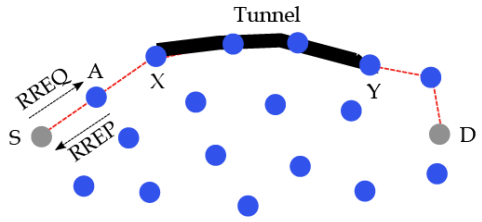

Fig. 1 – Combined attack: blackhole and wormhole attacks

III. Collaborative Detection and Defense in Ad Hoc Networks

Considering that the IDS in place has been able to detect a collaborative attack underway, the appropriate response needs to take place immediately. Assuming the security applications in the nodes are collaborating, various issues arise. There may be a central entity to collect data from each node on the network and take actions against intruders. Nodes may exchange data about attacks but act on their own. The processes involved in the attack may be closed, killed, shutdown, etc. The intruder or intruders may be blocked. The contaminated host may be isolated [8], and so on. These are all design features that have to be taken into consideration toward a robust response system.

We describe next our initial thoughts on devising a robust security system to not only detect intrusion but also react properly to them.

A. *Detection Mechanism*

Malicious nodes should be identified and prevented from communicating in the network. Our preliminary scheme to identify intruders comprises two layers: anomalous detection layer, using wavelet transforms [15],[16],[17],[18],[19], and classification of anomaly layer, using fuzzy logic [20][21],[22].

1) Anomalous detection with wavelet transforms

The wavelet transform is a mathematical technique capable of performing functions decomposition, and has been largely used as a signal processing tool in telecommunications; as well as in other fields. Using this technique it is possible to detect anomalies inside the network. The wavelet keeps track of parameters such as the network traffic behavior and number of connection

attempts, and can announce unexpected variations in such parameters. Because of that, many researchers have been investigating the feasibility of using wavelets in IDSs [17],[18].

The wavelet transforms allow for decomposing a given signal in frequency and time domain, so the signal's main frequency components can be identified in a timeline. Wavelets use a prototype function called mother wavelet which has mean value equal to zero and is too sensitive to changes in the input signal.

Mathematically, the Continuous Wavelet Transform (CWT) is represented as shown in (1). The variables "a" and "b" correspond to the scale and offset parameters, respectively.

The function *f(t)*, discrete or continuous, is the one over which the wavelet transform is applied. In our scheme, this function is composed of the collected or measured features of the network.

$$CWT(\mathrm{a},\mathrm{b}) = \int_{-\infty}^{+\infty} f(t)\psi^{*}_{a,b}(t)dt \quad (1)$$

The function $\psi_{a,b}(t)$ is the wavelet itself, and it is defined as shown in (2). The asterisk indicates that it is the complex conjugate of the function. Wavelet transforms comprise various equations families that are used in many areas [24].

$$\psi_{a,b}(\mathrm{t}) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right), a \neq 0, \quad b \in \Re \quad (2)$$

Once chosen the wavelet $\psi_{a,b}(t)$ and the data set $f(t)$, the computed values through CWT(a,b) are denoted as the wavelet coefficients. Variations in the input data will be reflected in these coefficients. Hence, wavelets are able to detect subtle variations in the input signal (network data) and also to register the time the variations take place. These features render wavelets proper to detect anomalies in IDSs. The detected anomalies have to be mapped to potential threats, so countermeasures can be taken. We use the weight light fuzzy logic to carry out the mapping.

2) Classification of the Anomalies with fuzzy logic

Fuzzy logic is a superset of conventional (boolean) logic that has been extended to handle the concept of partial truth. It was first introduced by L. Zadeh in the 1960s [2] as a means to model the uncertainty of natural language, and has been widely used for supporting intelligent systems. A key feature of Fuzzy logic is to handle uncertainties and non-linearities, existing in physical systems, similarly to the reasoning conducted by human beings, which makes it very attractive for decision making systems.

A fuzzy logic system comprises basically three elements: A fuzzifier, an inference method (rules and reasoning) and a defuzzifier. Their roles are as follows.

Fuzzifier (toward Fuzzy Sets):

A fuzzifier is responsible for mapping discrete (also called crisp) input data into proper values in the fuzzy logic space. This is done by using membership functions (fuzzy sets) which may provide smooth transitions from false to true (0 to 1). Mathematically, a membership function associates each element μX(x) in the universe of discourse U with a number in the interval [0,1], as shown in (3):

$$\mu X : \mathrm{U} \rightarrow [0, 1] \quad (3)$$

Therefore, a fuzzifier maps crisp data $x \subset U$ into a fuzzy set $X \subset U$, and $\mu X(x)$ gives the degree of membership of x to the fuzzy set *X*, i.e., a real number in the range [0,1]. And here 1 denotes full membership and 0 denotes no membership. So, fuzzy sets are indeed an extension of the classical sets in which only full membership or no membership exist. Fuzzy sets, on the other hand, allow partial membership.

Fuzzy Rules and Fuzzy Reasoning:

Fuzzy systems perform reasoning on the input data by following a predefined inference method and fuzzy rules. The amount of rules depends on both the number of inputs and membership functions associated to each input. The general form of the l*th* fuzzy rule in the rulebase is:

$$R^{l} : \; if\ (x_1\ is\ F_1^{l})\ and\ (x_2\ is\ F_2^{l})\ and\ ...\ (x_p\ is\ F_p^{l}) \quad then\ (y\ is\ G^{l}) \quad (3)$$

Where F_k^{l} and G^{l} are fuzzy sets associated with the input and output fuzzy variables x_k and y, respectively, being $k = 1,...,p$. As an example of (2), we could have: if (temp. is high) and (humidity is high) then (room is hot).

Defuzzifier:

Once the input data have been numerically processed by fuzzy reasoning, they are converted back to crisp values. This task is performed by the deffuzzifier which combines mathematically the result of each rule into a single crisp value. There are several methods for doing so, and we use here the most widely used algorithm called gravity-of-mass (GOM) [22], which computes in the simplest case the weighted average over all rule outputs.

3) Implementation

From an implementation standpoint, the detection system in each node should work as illustrated in Fig. 2. The nodes run the wavelet algorithm with relevant input parameters such as throughput and port scan rate to detect anomalies around them. If an alarm is triggered, the transmitter will send an alarm packet to the coordinator node. This packet includes the most important parameters for the coordinator node to evaluate and classify the alarm.

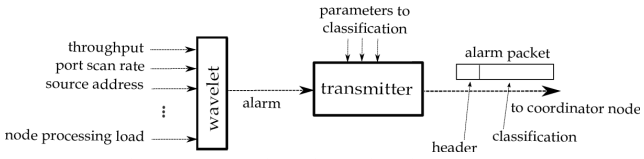

Fig. 2 – Intrusion detection in each node

B. Response Mechanism

An accurate detection is crucial for the response to be effective. That is, once the detection has been conducted

correctly, the response should be somewhat easier. A very important aspect of the response mechanism is to have an efficient way to exchange information among the nodes inside the network. This is important because of the bandwidth constraints inherent in these wireless networks.

Another crucial issue in responding mechanism regards the threshold upon which the response has to be triggered. This is also to be performed by the fuzzy logic-based mechanism responding to the intrusion.

We propose a semi-centralized approach, in which a node inside the network, coordinator node, is elected to be the aggregator of the alarms coming from the other nodes and also to be the source of the black list for the other nodes. It is semi-centralized because whenever a coordinator node leaves the network or is unable to communicate with the other nodes, a new coordinated node is elected. This proposed architecture not only reduces traffic overhead but most importantly also speeds up responsiveness.

Fig. 3 illustrates the proposed approach. Note that the anomaly detection is part of each node, so the coordinator node, N4, receives all the alarms the other nodes perceive. The classification and response tasks are done in the coordinator node.

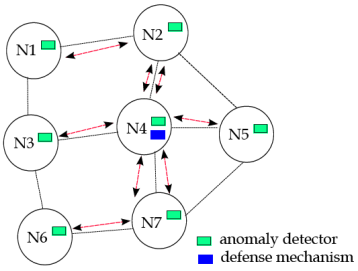

Fig. 3 – Semi-centralized approach

This approach is feasible in ad hoc networks used in semi-static environments such as office networks, campus networks, and library networks. In these environments, it is always possible to have a fixed node with wireless capabilities to serve as the coordinator node.

As a first attempt to implement the mechanisms involved in our approach, the coordinator node takes over the classification of the alarms and the defense strategy. All other nodes implement only the anomalous detection, using the wavelet-based algorithm. As said, the coordinator node is expected to be more powerful as far as both processing and energy capabilities are concerned.

Similarly to [5], the coordinator node gets all the incoming alerts from the other nodes, evaluates if there is an ongoing attack, and takes appropriate actions. The fuzzy logic-based mechanism at the coordinator node has two main tasks: to correlate the incoming alarms, and in case of a real attack if found to select the most effective response.

Using our proposed scheme, the fuzzy logic engine to be developed can infer that the collaboration shown in Fig. 1 is occurring and simply isolate node A from the network by putting it into a blacklist. As soon as the other nodes get the message from the coordinator node, the node A will be prevented from communicating with the network. In this particular scenario, the wormhole attack will not succeed, as it has no attractive route to the source node. Hence, this simple example underscores the importance of a collaborative response system.

An overview of the mechanisms in the coordinator node is shown in Fig. 4. The rx processor receives the alarm packets coming from the nodes and extracts the classification parameters to be handed over to the fuzzification mechanism. In this mechanism the classification parameters are mapped to predefined membership functions, being limited in the range 0-1. The reasoning mechanism processes the rules over the mapped values, and then the defuzzification algorithm computes the final output single value called crisp value. Depending on the output value, the system can infer that an intrusion is occurring and so put the detected intruders into a black list and send a broadcast message to other nodes.

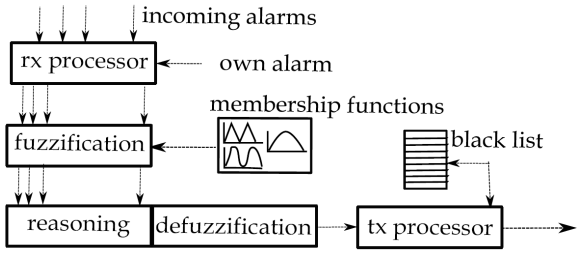

Fig. 4 – Intrusion detection and response at coordinator node

The efficiency of the proposed scheme depends strongly on the proper management of the black list. Once the coordinated node includes a given node in its black list, it has to keep monitoring the network and at some point in time decide whether if should or not remove the node from the black list. Since the nodes in the black list cannot rely on the well-behaved nodes to reach the coordinator node, the latter has to send specific messages to the isolated nodes for checking their current behavior.

By using this proposed mechanism it will be possible to promptly respond to collaborative attacks. For instance, if the response mechanism detects an ongoing attack, it may act quickly to protect the most valuable assets inside the network. This can be done by the coordinator node sending customized packets to some top priority nodes prior to the broadcast packet.

As long as the network mobility is low, what is true in various scenarios as said before, the proposed approach in Fig. 4 can work efficiently [20]. The main problem with high mobility is to find appropriate membership functions that represent well the movement-related changes in the network.

For designing the whole system in Fig. 4 various aspects have to be considered. In general, the involved parameters and their interrelationship have to understood and configured correctly. This will surely demand not only a broad theoretical analysis but also a well-designed testing approach using either simulations or real-life experimentations. In this work we only perform the initial phase of the many evaluations that are to be carried out toward the full-fledged secure approach. We evaluate in

the next section, the tradeoff between traffic overhead and responsiveness.

IV. PERFORMANE EVALUATIONS

The evaluations here were conducted using the Opnet Network Simulator [23]. The purpose of these simulations was to show that the proposed response system can be deployed in small-scale ad hoc networks without causing substantial performance degradations. This is a concern due to the traffic overhead resulting from the message exchanges involved in the semi-centralized approach.

We used the same topology as the one in Fig. 1 to simulate the effect of wormhole attacks on the performance of wireless networks. In this scenario, there are 19 nodes that are static and their transmission range is 250 meters. In these simulations we used DSR[24] as a routing protocol. All simulation runs lasted 1000 seconds, and to avoid disturbances from the warm-up period, the first 100 seconds of the simulation results were discarded.

Particularly, we simulated two different attack scenarios. In the first one we analyzed the effect of wormhole attack on the performance of ad hoc wireless networks, following the idea illustrated in Fig. 1. The second attack scenario was conducted to evaluate the power of collaborative attacks for a particular pattern of collaboration that can collapse the network.

A. *Impact of Reaction time on Halting Wormhole Attack*

For these evaluations, we divided the reaction time into two parts: the *detection time* and the *response time*. The former is the elapsed interval since the attack happens until the coordinator node receives the alarm from any node. The latter corresponds to the time the coordinator node spends to respond to the alarms and isolate the attackers if any real attack is indeed detected.

In order to improve the accuracy of the results, multiple, repeated attacks were conducted to each experiment. For the whole simulation time, the attacks were repeated each 200 seconds. Then the average of the measured values was calculated. And in each attack interval, 5 distinct reaction times were evaluated, namely 10, 20, 30, 40, and 50 seconds.

In these experiments, we evaluated four important metrics: *packet delivery ratio* (PDR), *normalized load* (shown here by the intrinsic overhead or simply OH), *average end-to-end delay*, and *throughput*.

PDR is computed by the ratio between the amount of packet delivered at the destination node and the whole amount of sent packets by the source node. OH represents the fraction of all control packets sent during the simulation time out of the total amount of packets transmitted, including data and control packets. By using OH, we can have a clear idea about how significant the overhead of the protocols and mechanisms on the network are. End-to-end delay and throughput complement the other two metrics in these experiments in the sense that they show from a different perspective the network performance degradation.

We were interested in seeing the impact of these attacks on the two most popular transport protocols: TCP and UDP. Hence, all the experiments were conducted for these two protocols, as shown in the figures below.

Fig. 4 shows the measured packet delivery ratio (PDR). As expected, by increasing the reaction time, PDR decreases for both TCP and UDP. This is a result of the large amount of dropped packets in the wormhole. The longer the reaction time, the more packets will be dropped in the attacked path. As the reaction time increases, the PDR can drop really substantially, as shown in Fig. 4.

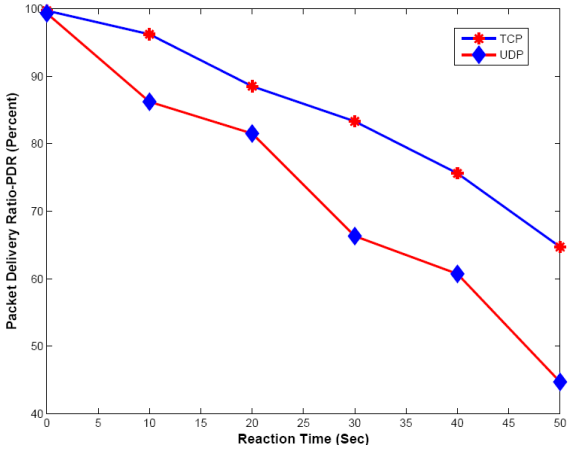

Fig. 4 – Packet delivery ratio under attack

It is interesting to note that TCP outperformed UDP in this experiment. Intuitively, TCP was expected to perform worst, as it uses a rate control mechanism that prevents the sender node from sending larger amount of packets without an acknowledgment from the receiver node. A possible explanation is that the conservative strategy of TCP in sending packets leads the system to lose fewer packets. For clarifying this issue we also measured the throughput of the connection between source and destination, and the results are shown in Fig. 5. By this outcome, it is proper to affirm that UDP provides higher throughput but also loses more packets, and because of that it obtains less PDR.

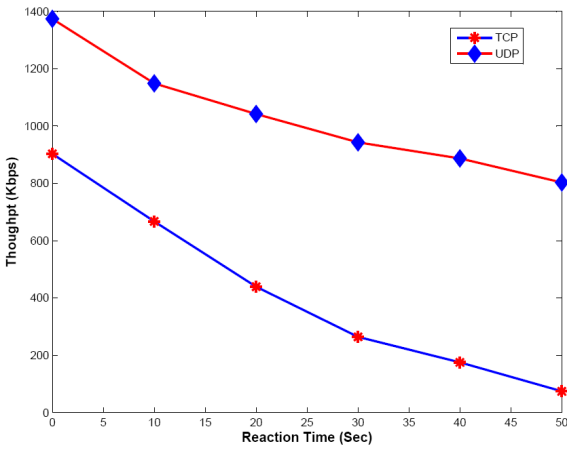

Fig. 5 – End-to-end throughput

The next metric evaluated was the network overhead. This metric shows us how much of control packets are generated, from detection to reaction, within the network. Again here the results in Fig. 6 illustrate the steadily

growth for higher reaction delay. TCP performs worse than UDP because of its rate control algorithm. Overall, as the losses raise the overhead increases as well. This is due the fact that the routing protocol in place has to exchange many control packets to establish new routes connecting sender and receiver. So the lower the reaction time, the better overall performance of our proposed scheme will be.

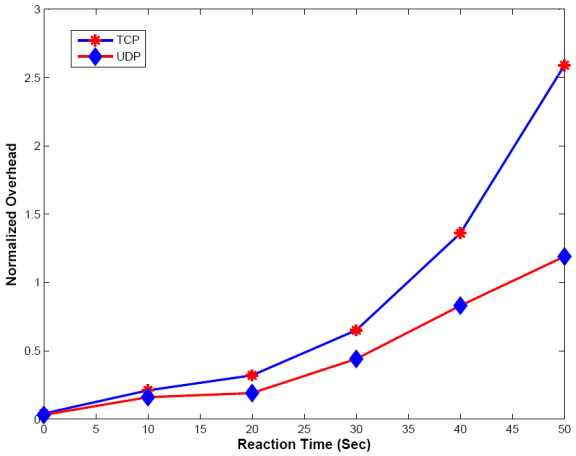

Fig. 6 – Overall overhead

The last observed metric, the end-to-end delay, is shown in Fig. 7. The results in this figure were calculated by taking the average of the end-to-end delay of the incoming packets at the receiver. As the previous metrics, the end-to-end delay stresses the importance of low reaction time by the defending mechanism.

.

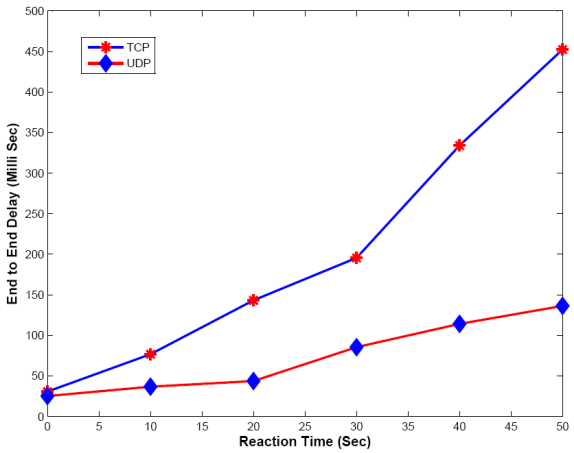

Fig. 7 – End-to-end delay

B. *Impact of Reaction Time to a Specific Collaborative Attack Pattern*

We conducted an experiment to show how specific attack patterns can be disastrous to the network. In this experiment, three nodes are deliberately collaborating to conduct a wormhole attack. Despite the simplicity of the collaborative scheme proposed here, the experiment was intended to highlight the problem that the learning techniques may face if they delay too much to recognize ongoing attacks. We assumed that the coordinator node takes on average 20 seconds to detect a malicious node.

The simple collaborative scheme works as follows: the three malicious nodes take turns in the attacks. That is, the first malicious node attacks for 7 seconds, then selects randomly the next attackers and passes the attack control to it. After 7 seconds, a new attacker is again randomly chosen, and so on.

The measured PDR in this experiment was completely annihilated, i.e., no packet got through. This means that even after the source node detects the first attacker and changes the route to reach the destination node, another attacker will be selected in the middle of the newly selected path. Hence, unless the defense mechanism is fast enough, the collaborative attack go undetected. Once again, this outcome underlines the importance of a fast response system. Our proposed scheme needs surely to take such cases into account.

C. *Discussions*

Overall, the simulations results allow affirming that the reaction time of the response mechanism may lead the network to collapse. In general, TCP is the protocol that will experience more degradation given its acknowledgment-based feature. But the results for the PDR evaluations showed that the TCP conservative nature may render it better in some scenarios. Since the results exhibit relatively good results to reaction delays up to 10 seconds, we are confident that our proposed scheme will work all for small scale ad hoc networks.

Based on that, for the upcoming experiments we intend to compare the effective delays experienced by our proposed scheme with the ones we measured here. This will be decisive for the designing parameters that will be part of our mechanism. Once our whole proposed mechanism is designed, the well-known false positive and false negative metrics will surely be evaluated.

V. *CONCLUSIONS*

We have examined important aspects of collaborative attacks and defense in ad hoc networks. In particular, techniques for collaboratively responding to intrusion in these networks have been proposed. We have elaborated on the way wavelet transforms and fuzzy logic algorithms can be used to defend low mobility ad hoc networks effectively. Relevant implementation issues such as the parameters to be used in each algorithm, the involved traffic overhead, and the responsiveness of the system, were addressed as well.

The preliminary evaluations showed that collaborative attacks can disrupt the network severely. The reaction time of the defense mechanism has to be bound to an acceptable level to guarantee efficient responsiveness. High reaction time will affect TCP-based application rigorously. Hence, all these aspects have to be considered in the design of the complete defense system proposed here.

For future work, it is interesting to test various parameters configurations for the techniques proposed in this work. This includes the selection of the most efficient

membership functions in the fuzzy logic-based mechanism. The capability of the response mechanism is also to be conducted. Evaluations of key issues such as efficiency, computation complexity, and energy consumption are also left for future work.

ACKNOWLEDGMENTS

This material is based in part upon work supported by the U.S. Department of Homeland Security under Grant Award Number 2006-CS-001-000001, under the auspices of the Institute for Information Infrastructure Protection (I3P) research program. The I3P is managed by Dartmouth College. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security, the I3P, or Dartmouth College.

REFERENCES

[1] A. Patwardhan, J. Parker, A. Joshi, A. Karygiannis and M. Iorga. "Secure Routing and Intrusion Detection in Ad Hoc Networks," Proc. third IEEE International Conference on Pervasive Computing and Communications, Mar. 2005.

[2] L. Qian, N. Song and X. Li, "Detecting and locating wormhole attacks in wireless ad hoc networks through statistical analysis of multi-path," IEEE Wireless Communications and Networking Conference (WCNC), Mar. 2005.

[3] Bharat Bhargava, "Intruder Idenfification in Ad Hoc Networks," CERIAS Security Center and Department of Computer Sciencies, Purdue University, research proposal 2002.

[4] B.Bhargava, R. Oliveira, U. Zhang, N. C. Idika, "Addressing Collaborative Attacks and Defense in Ad Hoc Wireless Networks," In: Second Workshop on Specialized Ad Hoc Network Systems (SAHNS 2009), in conjunction with the ICDCS, Montreal. 2009.

[5] F. Cuppens, A. Miege, "Alert correlation in a cooperative intrusion detection framework," Proc. IEEE Symposium on Security and Privacy, Toulouse, France, 2002.

[6] W. Lin, L. Xiang, D. Pao, B. Liu, "Collaborative Distributed Intrusion Detection System," fgcn, vol. 1, pp.172-177, 2008 Second International Conference on Future Generation Communication and Networking, 2008

[7] W. Yu-Sung, B. Foo, Y. Mei, S. Bagchi, "Collaborative intrusion detection system (CIDS): a framework for accurate and efficient IDS," Proc. Computer Security Applications Conference (ACSAC '03), 2003.

[8] N. Stakhanova, S. Basu, J. Wong, "A taxonomy of intrusion response systems," Int. J. Information and Computer Security, Vol. 1, No. 1/2, pp. 169-184, 2007

[9] S. Ramaswamy, H. Fu, and K. Nygard, "Effect of Cooperative Black Hole Attack on MANETs," Proc. ICWN, Jun. 2005.

[10] W. Wang, B. Bhargava, Y. Lu, and X. Wu, "Defending against Wormhole Attacks in Mobile Ad Hoc Networks," WCMC, vol. 6, issue 4, pp. 483-503, Jun. 2006.

[11] R. Maheshwari, J. Gao, S. R. Das, Detecting Wormhole Attacks in Wireless Networks Using Connectivity Information, Proc. of the 26th Annual IEEE INFOCOM'07, May, 2007.

[12] J. M. McCune, E. Shi, A. Perrig and M. K. Reiter, "Detection of Denial-of-Message Attacks on Sensor Network Broadcasts," Proc. IEEE Symposium on Security and Privacy, May 2005.

[13] H. Yu, M. Kaminsky, P. Gibbons and A. Flaxman, "SybilGuard: Defending Against Sybil Attacks via Social Networks." Proceedings of ACM SIGCOMM Conference, September 2006.

[14] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocol," 2nd ACM workshop on Wireless security, pp. 30-40, 2003.

[15] J. Gao, G. Hu, X. Yao, and R. C. Chang, "Anomaly Detection of Network Traffic Based on Wavelet Packet."In Proceedings of Asia-Pacific Conference on Communication, Aug. 2006.

[16] E. T. Ferreira, R. Oliveira, G. A. Carrijo, B. Bhragava, "Intrusion Detection in Wireless Mesh Networks Using a Hybrid Approach," In: Second Workshop on Specialized Ad Hoc Network Systems (SAHNS 2009), in conjunction with the ICDCS, Montreal. 2009.

[17] M. Hamdi, N. Boudriga, "Detecting Denial-of-Service attacks using the wavelet transform," Computer Comm. vol. 30, p. 10, 2007.

[18] C. Callegari, S. Giordano, and M. Pagano, "Application of Wavelet Packet Transform to Network Anomaly Detection," 2008, p. 246.

[19] M. H. Jansen, P. J. Oonincx, "Second Generation Wavelets and Applications. London," Springer, 2005.

[20] R. Oliveira and T. Braun, "A Delay-based Approach Using Fuzzy Logic to Improve TCP Error Detection in Ad Hoc Networks," IEEE Wireless Communications and Networking Conference (WCNC 2004), Atlanta, USA, March 21-25, 2004.

[21] L. A. Zadeh, Fuzzy logic = computing with words, IEEE Transactions on Fuzzy Systems, Vol. 4, No 2, pp. 104-111, 1996.

[22] L. Cheng and I. Marsic, Fuzzy Reasoning for Wireless Awareness, International Journal of Wireless Information Networks, Vol. 8, Issue 1, Jan. 2001, pp. 15-26.

[23] Opnet network simulator. http://www.opnet.com/solutions/network_rd/modeler.html.

[24] D. Johnson, D Maltz, J Broch, "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks", in Ad Hoc Networking, Chapter 5, pp. 139-172, Addison-Wesley, 2001