Using a Cache Scheme to Detect Misbehaving Nodes in Mobile Ad-Hoc Networks

Hongxun Liu, José G. Delgado-Frias, and Sirisha Medidi
School of Electrical Engineering and Computer Science
Washington State University
Pullman, WA 99164-2752, USA
{hliu2, jdelgado, smedidi}@eecs.wsu.edu

Abstract -- This paper presents a hardware based cache scheme to detect misbehaving nodes in mobile ad hoc network. In this scheme, the hardware monitors the activities of the upper-layer software and reports the misbehavior of the software to other mobile nodes in the network. The hardware cache stores the identity information of recently received packets. The detection mechanism uses the cache to detect packet dropping and packet misrouting. The simulation results show that the cache scheme can detect nearly 100% misbehaving nodes with nearly 0% false positive in the packet dropping scenario. In the packet misrouting scenario, the detection has nearly 0% false positive and more than 90% detection rate. The detection result could be used by other nodes to protect the network.

I. INTRODUCTION

Mobile ad hoc network (MANET) is a topic that has generated great research interest in recent years. Unlike traditional network, MANET does not use an infrastructure to work. Due to the features of self-configuration and self-maintenance, MANETs are usually deployed where there is no much communication infrastructures, for example, battlefields and disaster relief [1]. As the cost of the wireless devices drops, more and more peer-to-peer usage of MANETs are appearing; these networks are replacing fixed connections for some casual usage.

Some routing protocols for MANETs such as DSDV [2], DSR [3], and AODV [4] do not take into consideration security issues, assuming all the mobile nodes in the MANET will coordinate with each other. Such routing protocols along with the absence of infrastructure make MANETs vulnerable to active and passive attacks. For example, selfish nodes may drop packets silently to save energy. Dropping of packets may also be due to mobile node's software being compromised by a virus. The compromised software could show some misbehaving symptoms without the knowledge of the user. The existence of misbehaving nodes is a significant problem in MANETs; these nodes waste the wireless resources and degrade network performance. Having 10% to 40% misbehaving nodes degrades the average throughput by 16% to 32%, while a specific node could experience much worse performance degradation than the average [5]. Specifically, packet dropping can cause huge degradation to TCP based traffic due to the slow start nature of TCP [6].

Some techniques have been proposed to protect MANETs. For example, proactive techniques usually use encryption to protect MANETs by preventing attacks from happening. However, the security history shows that getting a completely intrusion-free system is not feasible, no matter how carefully the prevention systems are built [7]. Another issue about encryption based techniques is that it requires large computing power on the mobile node while usually the mobile node has a limited computing power. Some reactive techniques detect the misbehaving nodes and isolate the misbehaving nodes from the network [5]. A major issue about such techniques is that they tend to yield a number of false positive detections. .

In this paper a novel hardware assisted detection scheme is proposed and evaluated. In this scheme, the hardware can detect the misbehavior conducted by the misbehaving nodes. The misbehaving nodes could be selfish, or malicious. Selfish nodes are mobile nodes optimizing their own gain and neglecting for the welfare of other nodes. In this paper, the selfish node either drops all the packets not related to it or drops the data packets only. The malicious node misroutes the data packets to its colluding partner in stead of the real destination. The colluding node could analyze the traffic and gain valuable information. Upon detecting the misbehavior, the hardware will report the misbehaving node (itself) to other nodes. The other nodes will use the information received to protect the network. For example, other nodes could isolate the misbehaving nodes from the network. In this scheme, there is a separation between software and hardware inside a single mobile node. The software could be misbehaving, but the hardware is tamper resistant [8] and is the cornerstone of building trust relationship among mobile nodes.

The rest of the paper is organized as follows. Related work is reviewed in Section II. Then, Section III describes the hardware assisted cache scheme. Next, Sections IV and V describe the simulation and performance evaluation, respectively. Finally, conclusion and further research are discussed in Section VI.

II. RELATED WORK

Some techniques have been proposed to deal with misbehaving nodes in MANETs. The "watchdog" technique requires that each wireless node snoop other nodes on

retransmission of packets [5]. The watchdog raises a warning message if it finds out the other node didn't retransmit the packet correctly. Watchdog requires omni-directional antennas and cannot tell the difference between transmission error and misbehavior. So, watchdog could mistakenly indict some innocent nodes as misbehaving nodes. Furthermore, if a node is compromised, it can falsely accuse other nodes as misbehaving. Such false indictments are disastrous to the performance of MANET. In [5], there is no data available for the detection accuracy purpose. Buchegger and LeBoudec [9] propose a similar technique which requires encryption based associations among wireless nodes. The encryption work requires the wireless nodes having good calculation ability which is not conveniently available to wireless nodes due to battery usage. This technique can not prevent mobile node making false indictments. Papadimitrados and Haas [10] propose the Secure Routing Protocol (SRP), which extends DSR by adding SRP header to the basic routing header. The DSR packets are protected through a hash chain. SRP assumes there is already a security association between source node and destination node, which is difficult to establish in MANET. At the same time, there is short of a framework to accuse the node doubtlessly [11]. Paul and Westhoff [11] claim their hash-chain based framework can doubtlessly accuse a misbehaving node. But the voting scheme can only guarantee the accusation by some probability. Also, if the misbehaving node chooses to drop the packets silently, there is no way the neighbors can tell it is a misbehaving node or it doesn't even exist.

Stejano and Anderson [12] propose the "Resurrecting Duckling" model which builds security associations between master devices and slave devices. This technique simulates the phenomenon imprinting, where the device will recognize as its owner of the first device sending it a secret key [12]. Resurrecting Duckling focuses on the authentication among wireless nodes and can't detect the misbehaving nodes if the software of such nodes is compromised. Furthermore, the imprinting can only happen through close range transmission or by direct contact between master device and slave device.

The "Unobtrusive Monitoring" technique to detect packet dropping is proposed in [13]. In this scheme, all wireless nodes collect the TCP timeout events. If there is no corresponding DSR route error message or ICMP destination unreachable message to some TCP timeout event, such TCP timeout is taken as being caused by some misbehaving node. One issue with that scheme is that it can only tell that some wireless node along the TCP connection is behaving. It cannot pinpoint the specific node that is misbehaving. Another problem is the detection is conducted offline.

III. Hardware Assisted Detection

Nodes in MANETs may become selfish and not forward packets to preserve their battery life or misbehave if compromised by an adversary. It is very difficult to build a reliable trust relationship among mobile nodes without predefined security association. In this case, tamper-resistance hardware can act as the foundation of building reliable trust relationship among mobile nodes in MANET. Tamper resistant hardware is very popular in the smart card field [8]. When the software of a mobile node is tampered with or without the knowledge of user, the hardware can detect the misbehavior and send warning message to other nodes in MANET. After receiving the warning message, other mobile nodes can lower the rating of the misbehaving node or isolate the misbehaving node from the network. This paper focuses on the detection functionality of the hardware scheme.

A. Packet Dropping

Misbehaving nodes can either drop all the packets known as simple dropping or selectively drop packets. Typically misbehaving nodes selectively drop data packets and not the control packets such as route request, route reply, etc. because if they did they would not appear as a valid node in the paths collected by other nodes.

B. Packet Misrouting

In the MANET, a malicious node can misroute the data packets to its colluding partner. The colluding partner can then conduct traffic analysis and gain valuable information. For example, a malicious node can extract user name and password from the packets and use such information to gain monetary benefits. In this scenario, the malicious node misrouting the packets could be a normal mobile node, while the colluding node analyzing the traffic could be a high-performance node. Since the malicious will not gain useful information from the route request packet, it does not misroute route request packet. Furthermore, forwarding the route request packet correctly can help the malicious node receive the data packets followed. So, in this scenario, the malicious node only misroutes data packets.

C. Cache Scheme for Packet Dropping

In hardware assisted detection scheme, the hardware is responsible to detect the misbehavior of the software and report such misbehavior to other nodes. In the cache based detection scheme, there is a cache unit as well as a few counters. The cache stores the identity information of the recently received packets and is used to differentiate original packets from duplicate packets received by wireless node. Route discovery is done by broadcasting the request in mobile ad hoc networks; this could lead to broadcast storm problem resulting in the nodes receiving multiple route requests. To alleviate this problem, the cache can assist the detection hardware distinguish between original route request from the duplicate route request and not penalize the nodes for dropping the duplicate request.

There are four counters used in the cache based detection of packet dropping:

TC Total Counter
DC Drop Counter
DC Total Data Counter
DDC Data Drop Counter

The first two counters are used to detect simple dropping while TDC and DDC are used to detect selective dropping. TC is used to record the total number of unique packets received, while DC is used to record how many unique packets are dropped by this node. TDC is used to record how many data packets are received by the node while DDC records the number of data packets dropped.

The processing of incoming route request packet is depicted in Figure 1. As shown in Figure 1, each cache item has four fields: valid flag, source address, destination address and request number. When the detection hardware receives a route request, it will query the cache table if the same route request has been received before by using the source address and destination address as index. If there is a match in the cache table and the valid flag is set, the request number of the incoming packet is compared with the request number stored in the cache table. If the request number of the incoming route request is larger than the one in the cache table, the route request is original. Other wise, the incoming route request is a duplicate request. If the incoming route request is original, the cache table is updated with the new route request and the counters TC and DC are increased by 1. When the cache table is full, there needs a replacement strategy to find a slot for the new item in the cache table. The replacement strategy applied is first-in-first-out, a simple and effective algorithm.

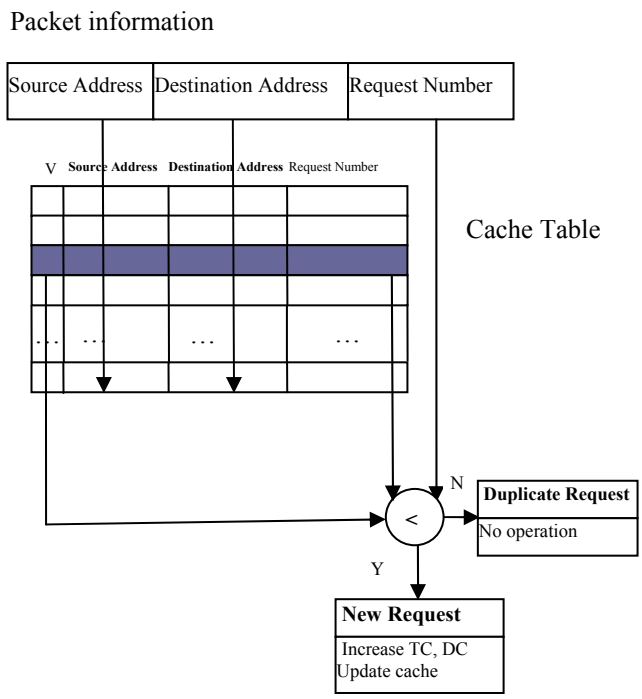

Figure 1. Processing of incoming Route Request Packet

If the incoming packet is a data packet, there is no need to query the cache table. In this case, the counters TDC and DDC are increased by 1.

For each outgoing packet, the counters are processed according to the type of packet received. If the outgoing packet is a route request packet, DC is decreased by 1. If the outgoing packet is a data packet, DDC is decreased by 1.

At some point, if either the ratio of DC to TC, or the ratio of DDC to TDC exceeds some predetermined thresholds, the detection hardware will send out a warning message to other nodes or the upper layer of the node. To decrease the false positive, DDC and DC need to be larger than some base value before the node could be indicted as misbehaving. The other nodes receiving the warning message could lower the rating of the node indicted or even isolate the node.

The basic cache scheme described above needs another timer to improve the detection performance. The timer is used to give additional penalty if a node doesn't forward a route request. The penalty timer (PeT) is started when an original route request is received. If the node doesn't forward the route request during the period of PeT, an extra penalty is added to DC. PeT is only started when an original route request is received. A duplicate route request will not initiate PeT. The cache unit inside the detection hardware can tell if a received route request is original or duplicate.

The purpose of PeT is to give additional penalty for dropping original route request packet. Thus good-behaving nodes and misbehaving nodes will have more obvious differences in terms of the values of Drop Counter (DC). Such additional penalty can improve the detection performance in terms of detection effectiveness and false positive.

D. Cache Scheme for Packet Misrouting

In order to detect packet misrouting, the cache unit needs to contain the identity of the data packet and the route used by the data packet. Besides the identity and route information, there is also a valid flag for each cache item. There is another counter, Data Misrouting Counter (DMC), used to detect the misrouting behavior.

When the detection hardware receives an incoming data packet, it will check if the node itself is the destination. If it is true, the hardware does nothing but give the packet to the upper layer. Otherwise, the hardware will query the cache unit if the same data packet has been received before. If the incoming data packet is a new one, a new cache item containing the identity and route of the packet is inserted into the cache table. If the data packet is a duplicate, there will be no update to the cache table. On the other hand, if the data packet is original, the identity and the path of the packet are inserted into the cache table along with DMC increased by 1.

For an outgoing data packet, the hardware detection unit checks if the source is the node itself. If the data packet is generated by the node itself, the hardware does nothing but send the packet out through the physical channel. Otherwise, the hardware will query the cache unit if the there is a hit for this packet. If there is a hit, DMC is decreased by 1. Otherwise, the hardware does nothing but send out the packet through the physical channel.

If the ratio of MDC to TDC is larger than a predetermined threshold, the detection hardware will send out a warning message about the misrouting behavior of the node. Thus, the other nodes receiving the warning message can exclude the indicted node from the network.

IV. SIMULATIONS

We implemented our techniques using the ns-2 simulator. Experiments were conducted using DSR routing protocol; the techniques can be easily extended to work with other routing protocols such as AODV.

A. Simulation Scenario

The simulations were conducted with 50 nodes in a 670 (meters) x 670 (meters) topology. Since our techniques are specifically designed to work with mobile ad hoc networks, randomly generated mobile networks were simulated conforming to the *random waypoint mobility* model [14]. In this model nodes choose a destination and move in a straight line toward the destination at a speed uniformly distributed between 0 meters/second and some maximum speed. With a maximum speed of 20 meters/second and a pause time of 5 seconds, we used 20 CBR connections. Each CBR connection generates packets using the interval of 1 second. The simulation length is 200 seconds.

B. Misbehaving Nodes

To study the proposed algorithms in a more hostile environment, nodes were configured to misbehave in terms of simple dropping, selective dropping, or misrouting. Nodes were randomly configured to be misbehaving. To notice the effectiveness of our approach, the percentage of misbehaving nodes was varied from 0% to 40% in 10% increment.

C. Detection Effectiveness and False Positive

The objective of the simulation is to show if the cache scheme can detect the misbehaving nodes accurately and efficiently. The primary metrics for evaluating the performance of our techniques are *detection effectiveness* and *false positives*. Thus two evaluation metrics are used in the evaluation: false positive and detection effectiveness.

Detection Effectiveness measures how well the cache scheme performs in identifying the misbehaving nodes. For example, if the scheme detects all of the misbehaving nodes in the network, the detection effectiveness is 100%. On the other hand, if it cannot detect any misbehaving node, the detection effectiveness is 0%.

Reporting misbehavior when none occurred is called a "false positive." The perfect detection would have 0% false positive and 100% detection effectiveness. But this in turn is extremely difficult to obtain. Our goal in this study is to keep false positive as low as possible and try to get the detection effectiveness as high as possible. Having a number of *good* nodes being misclassified has a negative effect on the overall performance of the network. On the other hand, some nodes that have the potential of misbehaving may not be involved in any communication path. This in turn makes those nodes difficult to detect. At the same time such remote nodes have minimum impact on the network performance because they are not engaged in valid data communication. Our first goal is to get very low false positive, followed by high detection effectiveness.

D. Cache

The cache unit inside the detection hardware can differentiate the original packets from the duplicate packets. It stores the identity information of the recently received packets. There are two cache units used in the detection hardware: one is used to detected packet dropping; the other one is used to detect packet misrouting. Because it needs not to store the whole packet but the identity information, the size of each cache item is small.

The number of cache items is expected to affect the detection performance in terms of detection effectiveness and false positive. On one hand, if the number of cache items is too small, the cache will misidentify some duplicate packets as original packets due to the quick replacement of the old items. On the other hand, too many cache items waste resource without gaining significant performance improvements. The simulations will find the optimal number of cache items.

E. Simulations

We conducted three sets of simulations to evaluate the performance of our techniques. Simulation set 1 (SS1) is used to detect simple dropping. Simulation set 2 (SS2) is used to detect selective dropping. The third simulation set (SS3) is used to detect packet misrouting. As mentioned earlier, we varied the percentage of misbehaving nodes for 10% to 40% in increment of 10%. Each data point represents an average of 20 simulation runs with different sets of misbehaving nodes.

V. PERFORMANCE EVALUATION

The performance of the cache assisted detection is evaluated through three different scenarios: simple dropping, selective dropping, and packet misrouting. The metrics used are false positive and detection effectiveness.

A. Simple Dropping Scenario

The set of simulations conducted in this section is to show the detection performance of cache scheme in the simple dropping scenario. In this scenario, the misbehaving node will drop every incoming packet if that packet is neither from itself nor to itself. Since the misbehaving node will drop all the incoming route request packets, it will not receive valid data packet to be forwarded.

Figure 2 shows the false positive using different cache sizes. The false positive decreases as the size of cache increases. This trend holds for all the four percentages of misbehaving nodes. When the size of cache increases to 8, the false positive is 0%.

After this point, the increment of cache size will not gain further significant benefit. The false positive stays steadily at 0% when the cache size is at least 8.

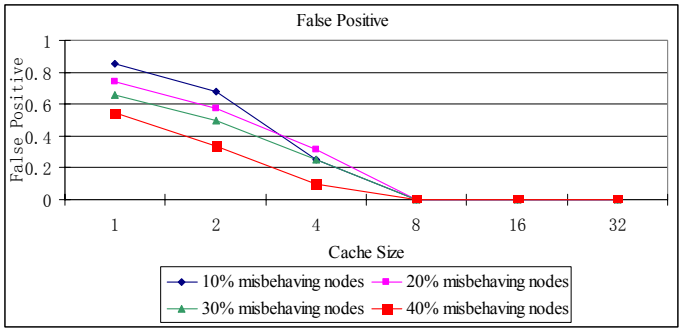

Figure 2. False positive in simple dropping scenario

In Figure 3, the detection effectiveness does not change a lot when the cache size increases from 1 to 32. When the cache size is 1, the detection effectiveness is at least 98%, of course with a lot of good-having nodes misidentified as misbehaving. As the cache size increases, the detection effectiveness decreases just a little bit. When the cache size reaches 32, the detection effectiveness is still at least 96%.

We have mentioned that our first goal is to keep the false positive as low as possible while keeping the detection effectiveness relatively high. According to the discussion above, as the size of cache increases there is no significant change in the detection effectiveness. So we can mainly focus on false positive. The false positive reduces to almost 0% when the cache size is equal to or larger than 8. Since there is no significant gain when the cache size is larger than 8, we could decide that 8 cache items are enough in the cache scheme proposed. Also, if we want more tolerance for different data traffic, we could choose to use 16 cache items.

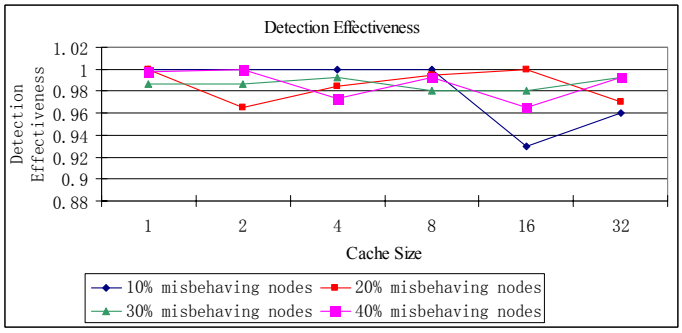

Figure 3. Detection effectiveness in simple dropping scenario

B. *Selective Dropping Scenario*

In selective dropping scenario, the misbehaving node will drop every incoming data packet unless that packet is from itself or to itself. At the same time, the node will forward control packets, such as route request, route reply, etc. The cache size is set as 8 which is confirmed in the simple dropping scenario above.

Table 1 shows the false positive and the detection effectiveness for different percentages of misbehaving nodes. From the results shown in Table 1, the cache scheme has almost 0% false positive and nearly 100% detection effectiveness.

TABLE 1. FALSE POSITIVE AND DETECTION EFFECTIVENESS OF ONLY COUNTING NODES COMMITTING DATA PACKETS DROPING

Parameter	10% misbehaving nodes	20% misbehaving nodes	30% misbehaving nodes	40% misbehaving nodes
False Positive	0%	0%	0%	0%
Detection Effectiveness	100%	100%	99.5%	100%

C. *Packet Misrouting Scenario*

In the packet misrouting scenario, the misbehaving node will misroute every incoming data packet unless that packet is from itself or to itself. At the same time, the node will forward control packets, such as route request, route reply, etc. Since misrouting detection uses a different cache unit from the one used in the packet dropping detection, we need to determine the optimum size of the misrouting detection cache.

Table 2 shows the simulation results for different sizes of cache in the detection hardware. FP and DE are referring to False Positive and Detection Effectiveness separately. For all the cache sizes from 1 to 8, the false positive is 0%. On the other hand, the detection effectiveness is almost the same for all sizes of cache. As mentioned before, 20 simulations are run for each cache size. One contribution to the same detection effectiveness is that the same random series is used for different cache sizes. Since there is no noticeable change among different sizes of cache, it is enough to use a cache unit with small number of items. The other simulations followed use cache size of 2.

TABLE 2. FALSE POSITIVE AND DETECTION EFFECTIVENESS FOR DIFFERENT SIZES OF CACHE ITEMS

Cache Size (entries)	10% Misbehaving Nodes		20% Misbehaving Nodes		30% Misbehaving Nodes		40% Misbehaving Nodes	
	FP	DE	FP	DE	FP	DE	FP	DE
1	0%	86.68%	0%	90.84%	0%	90.84%	0%	92.48%
2	0%	86.68%	0%	90.84%	0%	90.53%	0%	92.38%
4	0%	86.68%	0%	90.84%	0%	90.53%	0%	92.38%
8	0%	86.68%	0%	90.84%	0%	90.53%	0%	92.38%

After showing the effectiveness of the detection scheme, another set of simulations is run to determine the detection ratio used to trigger the warning message. On one hand, a small ratio could lead to high detection effectiveness and high false positive. On the other hand, a big ratio could lead to low detection effectiveness and low false positive. Thus, an

appropriate ratio helps achieve both good detection effectiveness and low false positive. If there is a conflict between high detection effectiveness and low false positive, the low false positive is preferred over detection effectiveness.

Table 3 gives out the simulation results of using different detection ratios. When detection ratio increases from 0.1 to 1, there is little change in detection effectiveness while maintaining almost 0% false positive. Thus, it is safe to say a relative low detection ration such as 0.1 or 0.2 would give both high detection effectiveness and low false positive.

TABLE 3. FALSE POSITIVE AND DETECTION EFFECTIVENESS FOR DIFFERENT DETECTION RATIOS

Detection Ratio	10% Misbehaving Nodes		20% Misbehaving Nodes		30% Misbehaving Nodes		40% Misbehaving Nodes	
	FP	DE	FP	DE	FP	DE	FP	DE
0.1	0%	91.67%	0%	91.04%	0%	92.50%	0%	95.02%
0.2	0%	91.67%	0%	91.04%	0%	92.50%	0%	94.46%
0.3	0%	91.67%	0%	91.04%	0%	92.50%	0%	93.29%
0.4	0%	91.67%	0%	90.04%	0%	92.50%	0%	92.58%
0.5	0%	91.67%	0%	90.04%	0%	92.50%	0%	91.40%
0.6	0%	91.67%	0%	90.04%	0%	92.50%	0%	91.40%
0.7	0%	91.67%	0%	90.04%	0%	91.94%	0%	91.40%
0.8	0%	91.67%	0%	90.04%	0%	91.94%	0%	90.95%
0.9	0%	91.67%	0%	88.79%	0%	91.38%	0%	90.95%
1	0%	91.67%	0%	88.79%	0%	91.38%	0%	90.95%

VI. CONCLUDING REMARKS

Mobile ad hoc networks are more vulnerable to misbehaving activities than the wired networks, which makes securing the mobile ad hoc networks very promising and enormously important. This paper presents a hardware based cache scheme to detect the misbehaving nodes. The features of the proposed schemes are:

- *High detection of misbehaving nodes.* The proposed scheme can detect nearly 100% misbehaving nodes in simple dropping scenario and selective dropping scenario. For packet misrouting scenario, the detection rate is almost 90%.
- *Zero false positive.* The cache scheme can achieve almost 0% misclassification of good-behaving nodes as misbehaving nodes in both packet dropping and packet misrouting scenarios.
- *Minor changes to software layer.* The proposed scheme requires very little change to the present software layer and can be easily implemented at the hardware layer due to the simple nature of the scheme.

The cache scheme can detect the misbehaving nodes accurately in terms of detection effectiveness and false positive in both packet dropping scenarios and packet misrouting scenarios. Currently, we are conducting some enhancements to the cache scheme and trying to combine the two cache units into one unit.

REFERENCES

[1] R. Rao and G. Kesidis, "Detecting malicious packet dropping using statistically regular traffic patterns in multihop wireless networks that are not bandwidth limited," *Proceedings of IEEE GLOBECOM*, pp 2957 – 2961, 2003.

[2] C. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *ACM SIGCOMM Computer Communication Review,* pp 234 – 244, October 1994.

[3] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[4] C. Perkins and E Royer, "Ad Hoc On-Demand Distance Vector Routing," *Second IEEE Workshop on Mobile Computer Systems and Applications,* pp. 90-100, February 1999.

[5] S. Marti, T. J. Guili, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," *Proceedings of ACM SIGCOMM*, pp 255–265, 2001.

[6] C. Barakat, E. Altman, and W. Dabbous, "TCP Performance in a heterogeneous Network: A survey," *IEEE Communication*, vol. 38, pp 40–46, January 2000.

[7] H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang, "Security In Mobile Ad Hoc Networks: Challenges And Solutions," *IEEE Wireless Communications*, pp 38-47, February 2004.

[8] S. Ravi, A. Raghunathan and S. Chakradhar, "Tamper Resistance Mechanisms for Secure Embedded Systems," *Proceedings of the 17th International Conference on VLSI Design*, 2004.

[9] S. Buchegger and J. Y. Le Boudec, "Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks," *Proceedings of the Parallel, Distributed and Network-Based Processing*, pp 403–410, January 2002.

[10] P. Papadimitrados and Z. Haas, "Secure Routing for Mobile Ad hoc Networks," *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002),* January 2002.

[11] K. Paul, D. Westhoff, "Context Aware Detection of Selfish Node in DSR based Ad-hoc Network," *IEEE GLOBECOM 2002,* November 2002.

[12] F. Stejano and R. Anderson, "The resurrecting duckling: Security Issues for Ad-hoc Wireless Networks," *Proceedings of the International Workshop on Security Protocols*, pp 172–194, April 1999.

[13] S. Medidi, M. Medidi, and S. Gavini, "Detecting Packet-dropping faults in Mobile ad-hoc networks," *Proceedings of IEEE Asilomar Conference on Signals, Systems and Computers,* pp 1708 – 1712, November 2003.

[14] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ah hoc network research," *Wireless Communication and Mobile Computing*, pp 483 – 502, 2002.