

CARE: Enhancing Denial-of-Service Resilience in Mobile Ad Hoc Networks

Ruiliang Chen, Jung-Min Park, and Michael Snow

Advanced Research in Information Assurance and Security (ARIAS) Lab
Bradley Department of Electrical and Computer Engineering
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061
{rlchen, jungmin, sno}@vt.edu

Abstract—This paper proposes an attack-resilient routing architecture, called Cross-layer Active RE-routing (CARE), for Mobile Ad hoc NETWORKS (MANETs). Different from existing solutions, CARE does not focus on a particular type of attack, but instead takes a fundamentally general approach—it achieves resilience against a wide range of routing disruption Denial-of-Service (DoS) attacks by treating them and “dysfunctional” network events in the same way. Here, dysfunctional network events denote link and routing failures caused by link contention or node mobility. CARE is a cross-layer scheme that detects attacks at the transport layer but responds to them at the network layer. Because dysfunctional network events and routing disruption attacks have a pronounced effect on the size of the TCP congestion window, monitoring the window size is an effective method of detecting such events. Using this method, CARE is able to detect attacks. Once an attack is detected, CARE initiates a re-routing process to find a new route. For this purpose, a re-routing algorithm is proposed that circumvents the nodes that are likely to be misbehaving. Analysis and simulation results show that the CARE architecture is effective in thwarting a number of insider and protocol-compliant attacks. Our results indicate that CARE is also effective in improving network throughput in non-hostile environments because its proactive re-routing mechanism aids in maintaining a reasonable level of throughput when dysfunctional network events occur.

I. INTRODUCTION

In a Mobile Ad hoc NETWORK (MANET), wireless devices communicate by forwarding packets on behalf of other devices—there is no central base station or fixed infrastructure to handle data routing. MANETs are particularly useful when a fixed infrastructure (e.g., a base station or access point) is impractical due to space or time constraints or when an existing infrastructure is not suitable for the required task. For mission-critical and other information-sensitive applications, the dependability and security aspects of the MANET, including reliability and availability, are of great importance. Denial of Service (DoS) attacks are a major threat to MANET security and quite a few of them have been discovered and discussed in the literature. Among them, routing disruption attacks are particularly menacing since they attempt to cause legitimate data packets to be routed in a dysfunctional way [5].

Routing disruption DoS attacks can be divided into three categories based on their different levels of sophistication: outsider attacks, insider attacks, and protocol-compliant attacks. In an outsider attack, the attackers are assumed to have no knowledge of the keys that are used to encrypt and authenticate the data and routing control packets. Preventing outside attackers from tampering with the data is accomplished by simply employing encryption and authentication schemes [5], [11].

In an insider attack, an attacker has compromised or captured a node, thus gaining access to encryption and authentication keys. The primary method of detecting and mitigating insider attacks is to monitor the packet forwarding behavior among the nodes [2], [3], [10], [12]. Also, there are approaches that focus on thwarting specific forms of insider attacks [6], [7].

Protocol-compliant DoS attacks [1] are the most difficult to defend against. In [1], Aad et al. refer to such attacks as “JellyFish” (JF) attacks. While the two types of attacks discussed above disobey protocol rules, JF attacks conform to all routing and forwarding rules. They are also passive, and therefore difficult to detect. A typical target of JF attacks is closed-loop flows that respond to packet delay and loss, such as TCP. Protecting MANETs against JF attacks is a formidable task that has yet to be addressed.

We propose a routing architecture for MANETs, called Cross-layer Active RE-routing (CARE), which is resilient against a wide range of attacks, including protocol-compliant attacks. CARE is a cross-layer approach that monitors the variations in the size of the TCP congestion window to detect abnormalities¹, and reacts to those abnormalities at the network layer by initiating a re-routing process. The CARE architecture is compatible with on-demand source routing protocols such as Dynamic Source Routing (DSR) [8]. CARE is composed of two modules: the *congestion window monitoring (CWM)* module and the *least-alike re-routing (LAR)* module. CWM is responsible for detecting any abnormalities that might occur on a route. If any abnormalities are detected, CWM invokes LAR to build a new route. Our simulation results show that CARE can effectively mitigate protocol-compliant attacks. Moreover, results indicate that CARE is capable of circumventing a variety of insider attacks.

The remainder of this paper is organized as follows. Section II provides the technical background of CARE. Section III introduces CARE and its modules. The security aspects of CARE are explored in Section IV. The simulation results are shown in Section V. Finally, we conclude the paper in Section VI.

II. TECHNICAL BACKGROUND

A. Related Work

Spoofing and *replay* are typical outsider attacks. They can be countered with encryption and packet authentication. Schemes

¹Network abnormalities can either be attacks or dysfunctional network events. The latter includes link and route failures that are caused by wireless link contention or node mobility.

that employ encryption or authentication include the Secure Routing Protocol (SRP) [11] and Ariadne [5]. The SRP attaches a security extension header to each control packet. SRP can be used for the DSR protocol. Ariadne [5] provides efficient authentication for DSR using a variant of the TESLA source authentication technique.

Various insider attacks have been discussed in the literature, including blackhole, grayhole [5], rushing [7], wormhole² [6], blackmail, and selfish attacks. The research community has made a great effort to combat insider attacks [2], [3], [6], [7], [10], [12]. Awerbuch et al. [2] introduced a technique for detecting faulty links on a path from the source to the destination using a binary search. Hu et al. proposed the *Rushing Attack Prevention (RAP)* scheme [7] as a generic defense against the rushing attack for protecting on-demand routing protocols. The same authors proposed “packet leashes” [6] to thwart wormhole attacks during a route search process. A reputation-based system is another approach that thwarts attacks by monitoring the network traffic [3], [10], [12]. For example, Marti et al. [10] proposed two modules—“watchdog” and “pathrater”—for this purpose. Watchdog is a module that detects neighbor nodes’ misbehavior in the promiscuous mode; and pathrater is a route selection module that defines a route’s quality as the average reputation of the nodes on the route and chooses the route with the best quality.

Protocol-compliant DoS attacks, a.k.a. JF attacks [1], is by far the most difficult to defend against. In a JF attack, the malicious node can reorder packets, periodically drop packets, or increase packet jitter. Although such behavior can be considered a network-layer attack, it affects the transport-layer goodput by exploiting the vulnerabilities of the congestion control mechanism. It was shown in [1] that the JF attack can result in near zero goodput in the transport layer while keeping network-layer throughput fairly stable. Currently, there is no known countermeasure for the JF attack³.

B. Overview of the Dynamic Source Routing Protocol (DSR)

CARE is an architecture for secure routing in MANETs that can be most readily integrated into on-demand source routing protocols. To describe CARE’s functionalities in a concrete way, we will discuss it in the context of an actual on-demand source routing protocol, namely DSR [8]. DSR uses the source routing option in data packets to carry the routing information. Each node, using a route cache, stores one or more complete lists of node addresses that form a path toward a destination. DSR is composed of two phases: route discovery and route maintenance.

Route discovery: When a node has packets to send, it first checks its route cache. If a route entry corresponding to the destination is not present in its route cache, a ROUTE REQUEST packet is broadcast over the network. The ROUTE REQUEST packet is uniquely identified by the source address, the destination address, and a sequence number that is incremented by the source node for each route discovery request. An intermediate node appends its own address to the node list in the ROUTE REQUEST packet and forwards it. When

²It can be argued that a malicious node in a rushing or wormhole attack could act like a repeater without requiring knowledge of any keys. However, to really do harm, the attacker needs to distinguish control packets from data packets in either of the attacks. This is possible only when the attacker has knowledge of the encryption key. Therefore, we classify both attacks as insider attacks.

³*L’Hospital* [9] uses a community-based secure routing to thwart JF attacks. However, this scheme assumes the existence of an intrusion detection system that is capable of detecting JF attacks.

the ROUTE REQUEST packet reaches the destination node, it has accumulated the path from the source to the destination. Assume that the underlying MAC layer supports bidirectional links, the destination node can get a valid route back to source node simply by reversing the source route recorded in the ROUTE REQUEST packet. Then the destination node sends back to the source node a ROUTE REPLY packet along the same route in the opposite direction. The ROUTE REPLY packet contains the information needed for the source node to route its packets to the destination node. It is possible for a node to receive the same ROUTE REQUEST packet multiple times because it is broadcast over the network. DSR requires an intermediate node to respond only to the first ROUTE REQUEST packet received and ignore the other duplicates, which is known as “duplicate suppression”. Note that duplicate suppression makes DSR vulnerable to rushing and wormhole attacks—if a malicious node manages to disseminate ROUTE REQUEST packets quickly so that they reach the other nodes before the legitimate packets, then the malicious node on a route with the least delay will always be included in the selected route.

Route maintenance: Every node along a route is responsible for the validity of the downstream link connecting itself and its next hop node. If link breakage is found, the source node will be notified with a ROUTE ERROR packet. The source node then initiates another route discovery procedure. Note that this procedure has a vulnerability: since sending a ROUTE ERROR packet is voluntary, malicious nodes can break links without being detected by the source node.

III. THE CROSS-LAYER ACTIVE RE-ROUTING ARCHITECTURE

A. Assumptions and Overview of CARE

We assume all links in the network to be bidirectional. We only consider network-layer route disruption attacks and disregard attacks to the physical or link layer of a wireless network. In a communication session, we assume that both the source node and the destination node are trustworthy but intermediate nodes are not. It is assumed that all control packets used in CARE are authenticated via certain security mechanism (e.g., [5], [11]). Under this assumption, CARE is inherently resistant to outsider attacks. We focus our discussions on insider attacks and protocol-compliant attacks. A route with one or more malicious nodes is considered an “infected” route.

In this paper, we focus on TCP as it is the most widely-used transport-layer protocol and it is the attack point of JF attacks. CARE uses a CWM module to observe the variations in the size of the TCP congestion window. If the variation indicates an abnormality, an alarm is raised to activate the LAR module. The LAR module in turn finds a new route. CARE strengthens the two vulnerabilities of DSR. First, CARE fortifies the “passive” re-routing approach of DSR by supporting both passive re-routing and a form of “active” re-routing. In DSR, the source node passively waits for a ROUTE ERROR packet to trigger a re-routing process. CARE enables the CWM module to actively initiate a re-routing process when network abnormalities are detected. Therefore, even if a malicious node on an infected route drops ROUTE ERROR packets, the source node is able to initiate re-routing. Second, CARE facilitates the process of identifying a valid route when a new route has to be found. The LAR module of CARE disables duplicate suppression when abnormalities are detected and considers route history in the re-routing process.

B. Congestion Window Monitoring (CWM)

Before giving the rationale behind the CWM module, we introduce the following theorem.

Theorem I: Suppose that \bar{W} denotes the mean value of the measured congestion window size in number of TCP segments, assuming that the TCP segment size is fixed. In addition, suppose that the congestion window size W at time t is represented by W_t , and the window size in the next RTT is given by W_{t+1} . Then the probability of W_{t+1} being less than or equal to W_t satisfies the following relation:

$$P[W_{t+1} \leq W_t] < \frac{8}{3\bar{W}^2 - 2\bar{W}}. \quad (1)$$

The proof of the theorem is given in the Appendix. In [4], Fu et al. conducted comprehensive experiments and simulations in order to observe TCP performance in various MANET environments differing in topology and traffic load. It was found that TCP maintains a relatively large average congestion window size. Specifically, the average measured window size \bar{W} was found to be between 12 and 26. We conducted similar but more extensive simulations⁴ and observed that $\bar{W} \geq 4$ for every simulation run. Applying Theorem I with $\bar{W} \geq 4$, we get $P[W_{t+1} \leq W_t] < \frac{1}{5}$. This probability is relatively small. Therefore, if we assume the events in which $W_{t+1} \leq W_t$ are independent, then it would be unlikely that multiple events will take place in a short time duration. Suppose there are consecutive K RTTs being monitored, and let q denote the number of RTTs with $W_{t+1} \leq W_t$. Given a constant $q_0 \in \{1, \dots, K\}$, the probability $P[q \geq q_0]$ is lower-bounded as follows:

$$P[q \geq q_0] < \sum_{i=q_0}^K \binom{K}{i} \left(\frac{1}{5}\right)^i \left(\frac{4}{5}\right)^{K-i}. \quad (2)$$

The summation of the above inequality is a very small value when q_0 is close to K . For example, when $q_0 = 8$ and $K = 10$, $P[q \geq q_0] < 7.8 \times 10^{-5}$. However, the value of $P[q \geq q_0]$ can be relatively large when a network is under a routing disruption attack. For instance, in a blackhole attack, an adversary node drops every data packet it receives. This prevents the TCP window size from increasing, and thus results in $P[q \geq q_0] = 1$. Even in low-frequency attacks, such as grayhole or Jellyfish attacks, the value of $P[q \geq q_0]$ may increase noticeably. Hence, it seems possible to use the value of $P[q \geq q_0]$ to detect the existence of attacks.

To verify this idea, we conducted extensive simulations using ns-2. We ran the simulations on a wide range of network topologies. Figs. 1 and 2 show the results obtained from two particular topologies: a 6-hop chain network and a randomly distributed 200-node network. The former and the latter represent respectively the most simple and the most complex topology among the ones that were considered⁵. A single TCP flow was assumed for the 6-hop chain network. The randomly distributed topology contains 200 nodes distributed in a 2000m \times 2000m area with ten TCP flows. We simulated a Periodic Drop JF attack—a subcategory of JF attacks—in which JF nodes forward all control packets and drop

⁴In particular, we simulated the chain, cross, grid, and random topologies as used in [4] but applied more diverse traffic loads, including those that exceed the network capacity.

⁵Due to space constraints, we have not included the simulation results of the other topologies whose complexity is somewhere in between the two considered above.

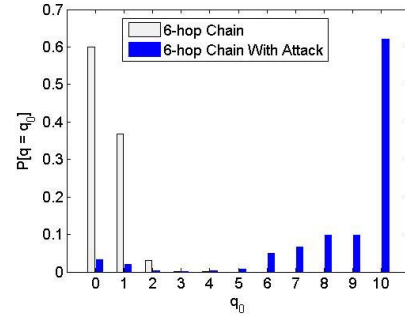


Fig. 1. Distribution of $P[q = q_0]$ for the 6-chain topology.

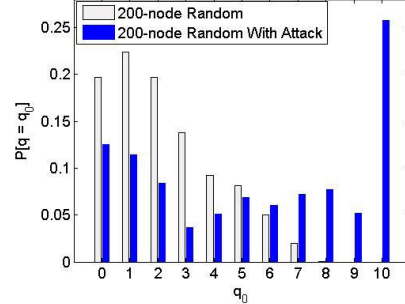


Fig. 2. Distribution of $P[q = q_0]$ for the 200-node random topology.

data packets for 100ms per second. JF nodes were placed as follows: a single JF node was placed in the middle of the chain for the 6-hop chain topology, and ten JF nodes were randomly distributed for the 200-node random topology. We used $K = 10$ and ran ten independent simulations for each topology. Although the severity of the attack on the two networks is moderate, it can be seen from Figs. 1 and 2 that it significantly affects the value of $P[q = q_0]$. For instance, one can observe from both figures that when $K = 10$ and $q_0 = 8$, $q \leq q_0$ occurs only in the attack scenario, and in that scenario, $q \leq q_0$ is observed for at least 38% of the total TCP session time. Therefore, the event $q \leq q_0$, which we refer to as the *stagnant window event* hereafter, serves as an effective indicator of attack existence.

Node mobility was not considered in the above simulations because we wanted to isolate the effect of the attacks. It will be considered when the CARE architecture is evaluated in Section V.

Not only attacks but dysfunctional network events (e.g., routing failures due to node mobility) may also cause the stagnant window event, thus incurring a “false positive”. CARE solves this problem by using the LAR module to build a new route for the current flow, irrelevant of whether the stagnant window event is due to an attack or a dysfunctional network event. In essence, CARE provides a general blanket of protection against a wide range of routing failures by responding to attack events and non-attack events in the same manner; this approach significantly simplifies the overall design of a secure routing protocol and avoids the use of “customized” security solutions whose effectiveness is limited to only certain attack types.

C. Least-alike Re-routing (LAR)

LAR enhances the re-routing functions of DSR in three aspects. First, when CWM detects any abnormalities in a

```

// Input:  $S_N = \{N_j : 1 \leq j \leq U\}$ ,  $S_R = \{R_i : 1 \leq i \leq M\}$ ;
// Output:  $j'$  such that  $N_{j'}$  is the result of running the LAR algorithm
int Least_alike(N[], R[]) {
    int E[U] = 0; // stores the alikeness degrees of  $\{N_j\}$ 
    int MIMA = 8, MINI = 8; // MIMA stores the minimum
                                // number of E[U]; MINI stores the
                                // minimum  $|N_j|$  with  $E[j] = \text{MIMA}$ 
    int result = 0; // used to hold the output value
    for (j = 1; j <= U; j++) {
        for (i = 1; i <= M; i++)
            E[j] = (E[j] > INTS(N[j], R[i])) ? E[j] : INTS(N[j], R[i]);
            // INTS(N[j], R[i]) is a function for  $|N_j \cap R_i|$ 
        MIMA = (MIMA < E[j]) ? MIMA : E[j];
    }
    for (j = 1; j <= U; j++) {
        if ((E[j] == MINI)
            && ((result == 0) || (LEN(N[j]) < LEN(result))))
            // LEN(N[j]) is a function for  $|N_j|$ 
            result = j;
    }
    return (result);
}

```

Fig. 3. C-style pseudocode of the LAR algorithm.

route, LAR tags the current route as an infected route. If there is no non-tagged route available in the source node's cache, LAR initiates an active re-routing process rather than just responding passively to ROUTE ERROR packets. Second, during the active re-routing process, the source node collects more routing information by disabling duplicate suppression⁶. Third, with the route information in its cache, the source node selects a new route using the following LAR algorithm.

Assume that the source node's set of cached uninfected routes are represented by $S_N = \{N_j : 1 \leq j \leq U\}$, where $U = |S_N|$ ($|X|$ denotes the number of elements in set X), and the routes that are tagged as infected are represented by $S_R = \{R_i : 1 \leq i \leq M\}$, where $M = |S_R|$. Here, N_j or R_i denotes a set of nodes contained in a given route. Let us define the "alikeness degree" of N_j with respect to S_R to be $E(j) = \max_i (|N_j \cap R_i|)$. Then, LAR algorithm can be expressed as follows: the sender selects a new route by selecting a route in the cache with the smallest alikeness degree. That is, it selects a new route $N_{j'}$ such that $j' = \arg \min_j (E(j))$. If there

are multiple routes that satisfy the aforementioned condition, then the one with the least number of nodes is selected among them. If there are still more than one routes to choose from, then the one with the smallest index is chosen.

It can be shown that for a k -hop route, if n nodes are inserted by an attacker and can be deployed anywhere, then the complexity of the LAR algorithm is $O(n^{k-1})$. Fig. 3 shows a C-style pseudocode of the LAR algorithm.

IV. SECURITY ANALYSIS OF CARE

CARE is designed to counter a wide range of attacks, including blackhole, grayhole, rushing, wormhole, and JF attacks. In this section, we analyze CARE in terms of its ability to defend against these attacks. We assume there is at least one uninfected route between a source node and a

destination node. We will refer to the example network shown in Fig. 4 in our discussions, where S and D are source and destination nodes, respectively, and A , B , C , E , F , and G are intermediate nodes.

Blackhole, Grayhole and JF attacks: In all three attacks, a malicious node attracts routes passing through it by forwarding control packets normally, but manipulate data packets once it is included in a route. A backhole drops all data packets while a grayhole drops data packets probabilistically. A JF attacker tampers with packets more artistically as described in Section II. If A is a blackhole/grayhole/JF node while other nodes are all benign, the TCP congestion window of any route containing A will be affected. Suppose that route $S-F-A-B-G-D$ has been selected and is currently in use. According to subsection III-B, if A , the malicious node, tampers with data packets, then CWM will readily observe the stagnant window event. Then LAR gets notified and the route will be tagged as being infected. Then, CARE will trigger a new route discovery with duplicate suppression disabled to find more routes, including $S-F-A-E-G-D$, $S-F-C-B-G-D$, and $S-F-C-E-G-D$. The LAR algorithm will output either $S-F-C-B-G$ or $S-F-C-E-G-D$, whichever one has the smaller route index.

Rushing attack: In a rushing attack, the malicious node suppresses ROUTE REQUEST packets forwarded by other nodes by disseminating ROUTE REQUESTs very quickly. If the malicious node forwards packets without any modification and makes itself transparent to the network layer, this behavior is called a repeater attack. In Fig. 4, assume that A is malicious while other nodes are all benign. Assuming that A has successfully launched a rushing attack, it can take two types of actions to disrupt routing. The first action A can take is not forwarding the ROUTE REPLY packet. However, this abnormality can be easily detected; S will send a new ROUTE REQUEST packet disabling duplicate suppression in response to no receipt of ROUTE REPLY. Then S will receive at least one ROUTE REPLY packet via routes that do not include A (e.g., $S-F-C-B-G-D$). The other action A can take is to forward ROUTE REPLY packets and place itself in the established route. Then A can launch a blackhole/grayhole/JF attack to disrupt communications between S and D . Again, this is ineffective since as discussed above, CARE counter the latter attacks.

Wormhole attack: In a wormhole attack, a pair of malicious nodes tunnel packets from one part of the network to another, thus disrupting routing by short circuiting the normal flow of routing packets. This attack can be regarded as a colluding rushing attack. Similar to rushing attacks, a wormhole attack is harmful only when launched together with blackhole/grayhole/JF attacks. Since CARE can detect and thwart these concomitant attacks, it can effectively counter harmful wormhole attacks. In Fig. 4, suppose that A and E compose a wormhole, and other nodes are all benign. Also, suppose that both A and E attempt to disrupt routing by launching a blackhole attack after a route has been established. Two scenarios are possible. In the first case, A and E add themselves to the list of route nodes and are seen by other nodes. Because A and E are blackhole nodes, route $S-F-A-E-G-D$ will be tagged as being infected by LAR, and the LAR algorithm will return $S-F-C-B-G-D$ as the next route to use, which is an uninfected route. In the second scenario, both A and E make themselves transparent to other nodes. Initially, route $S-F-(A)-(E)-G-D$ will be selected and used. Because A and E are blackhole nodes, CWM will detect a stagnant

⁶This can be implemented by adding a one-bit flag in the ROUTE REQUEST packet.

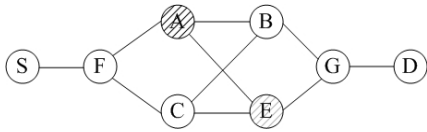


Fig. 4. An example network.

window event, and a re-routing process will be invoked. As a result, three new routes will be found: $S-F-(A)-B-G-D$, $S-F-C-(E)-G-D$, and $S-F-C-B-G-D$. All three routes have the same alikeness degree with respect to the routes recognized as being infected. Hence, either the first or second route will be selected, since they have shorter hop counts compared to the third route. Because both $S-F-(A)-B-G-D$ and $S-F-C-(E)-G-D$ contain a blackhole node, they will both be tagged as infected routes in the next two trials. In the final trial, the uninfected route $S-F-C-B-G-D$ will be selected.

V. SIMULATION STUDY

A. Simulation Environments

The simulation results shown in this section were obtained via ns-2.

We consider a network of 200 nodes in a $2000m \times 2000m$ square area. Nodes use the 802.11 MAC with a 250m communication range. Each node moves according to the random waypoint model, which repeats the following four steps:

- 1) It randomly chooses a destination in the area with a uniform distribution;
- 2) It chooses a velocity v that is uniformly distributed over $[v_{min}, v_{max}]$;
- 3) It moves along the straight line from its current position to the destination with the velocity v until it arrives; and
- 4) It pauses in the destination for a random period that is uniformly distributed over $[0, t_{max}]$.

We adopted the values $v_{min} = 10m/s$, $v_{max} = 20m/s$, and the $t_{max} = 10s$. With this model, ten different random movement patterns were generated. All of the results to be presented are averaged over five independent simulations on each of the ten movement patterns. We simulated 5-flow networks with 0, 16, 25, and 49 malicious nodes. The flows use TCP-NewReno senders with standard TCP receivers. Each flow sends packets at a rate of 2000 bytes per second. Malicious nodes launch a *Periodic Drop JF attack* in which JF nodes forward all control packets and drop data packets for 300ms in every one second interval. We chose the period of one second because the Periodic Drop JF attack with this period length was shown to have the most detrimental effect on TCP goodput according to the results published in [1]. The values used for the number of nodes, flows and malicious nodes are the same as those used in the simulation experiments of [1]. Each simulation run lasts 900 seconds.

B. Simulation Results

1) *Data Throughput and Control Overhead*: Figs. 5 and 6 compare DSR and CARE in terms of data throughput (i.e., goodput) and control overhead, respectively. Fig 5 shows that CARE achieves an increase in TCP goodput compared to DSR. Note that even when there is no malicious node, CARE increases the TCP goodput by 22%. This observation can be attributed to the fact that CARE actively conducts re-routing when node mobility causes link breakage, while DSR passively waits for ROUTE ERROR messages before initiating

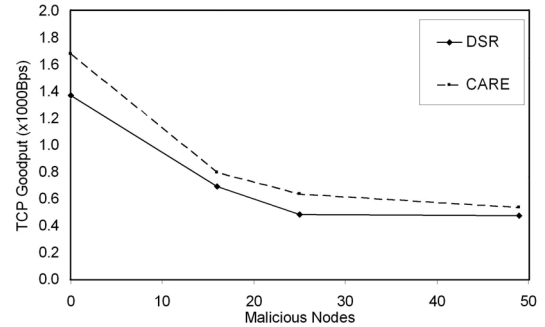


Fig. 5. TCP goodput.

re-routing. The goodput gain achieved by CARE decreases as the number of malicious nodes increases because the increase in the number of malicious nodes rapidly decreases available network resources. Fig. 6 shows that CARE pays the price of increased control traffic for the improvement in goodput—this can be attributed to the increase in route discovery attempts.

2) *Average Successful Route Length*: The route length averaged over all successfully transmitted packets is an indicator of a protocol’s ability to maintain multi-hop routes. Note that longer multi-hop routes are more vulnerable against routing disruption attacks. Fig. 7 shows that the average route length, measured in number of hops, is longer using CARE than using DSR under all scenarios, indicating that CARE is more resilient against attacks.

3) *System Fairness*: Jain’s fairness index, used in [1], is computed using long-term average throughput, and is a way of evaluating how well network bandwidth is shared. An index closer to one is desirable since it indicates that all flows receive an equal share of the network bandwidth. The averaged simulation results, shown in Fig. 8, indicate that CARE helps maintain better system fairness, which implies that CARE’s increased TCP goodput is distributed more heavily to the low-throughput routes than the high-throughput routes. This again shows that CARE is able to enhance the attack resilience of the network.

VI. CONCLUSION

This paper presented a novel routing architecture for MANETs called CARE that is attack resilient. CARE employs a cross-layer approach in that it uses the CWM module to detect network abnormalities (either attack or dysfunctional events) at the transport layer and responds to them by using the LAR module to execute re-routing at the network layer. Our analysis shows that CARE is resilient against a variety of insider attacks as well as protocol-compliant attacks. Simulation results show that CARE is effective in mitigating JF attacks in certain network environments. As part of our future work, we will explore the possibility of adapting the principles of CARE to routing protocols other than DSR.

REFERENCES

- [1] I. Aad, J. Hubaux, and E. W. Knightly, “Denial of service resilience in ad hoc networks,” *Proc. MobiCom*, Sep. 2004, pp. 202-215.
- [2] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens, “An on-demand secure routing protocol resilient to Byzantine failures,” *Proc. WiSe*, Sep. 2002, pp. 21-30.

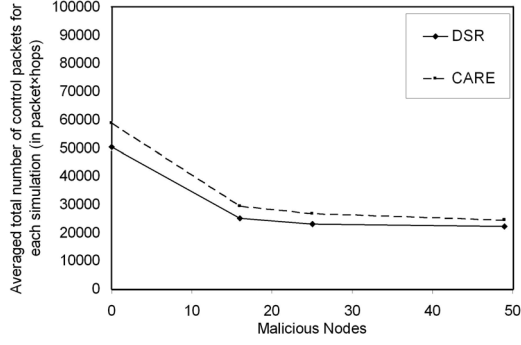


Fig. 6. Control overhead.

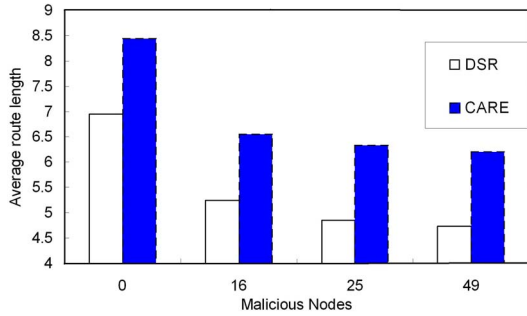


Fig. 7. Average route length.

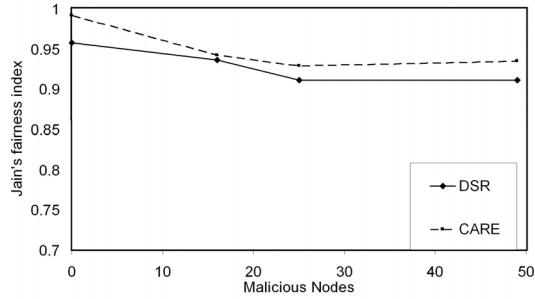


Fig. 8. Jain's fairness index.

- [3] S. Buchegger and J.-Y. Le Boudec, "Nodes bearing grudges: towards routing security, fairness, and robustness in mobile ad hoc networks," *Proc. 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing*, Jan. 2002, pp. 403-410.
- [4] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The impact of multihop wireless channel on TCP throughput and loss," *Proc. INFOCOM*, Mar. 2003, pp. 1744-1753.
- [5] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A Secure on-demand routing protocol for ad hoc networks", *Proc. MobiCom*, Sep. 2002, pp. 12-23.
- [6] Y.-C. Hu, A. Perrig, and D.B. Johnson, "Packet leashes: a defense against wormhole attacks in wireless networks," *Proc. INFOCOM*, Mar. 2003, pp.1976-1986.
- [7] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Rushing attacks and defense in wireless ad hoc network routing protocols," *Proc. WiSe*, Sep. 2003, pp. 30-40.
- [8] D. B. Johnson, D. A. Maltz, and Y.-C. Hu, *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR) (Internet-Draft)*, Mobile Ad-hoc Network (MANET) Working Group, IETF, July 2004.
- [9] J. Kong, X. Hong, J. Park, Y. Yi, M. Gerla, *L'Hospital: Self-healing Secure Routing for Mobile Ad-hoc Networks*, Technical Report TR-040055, Computer Science Department, University of California, Los

Angeles, Jan. 2005.

- [10] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," *Proc. MobiCom*, Aug. 2000, pp. 255-265.
- [11] P. Papadimitratos and Z. Haas, "Secure routing for mobile ad hoc networks," *Proc. CNDS*, Jan. 2002.
- [12] W. Yu, Y. Sun, and K. J. R. Liu, "HADOF: Defense against routing disruptions in mobile ad hoc networks," *Proc. INFOCOM*, Mar. 2005, pp. 1252-1261.

APPENDIX

Theorem I: Suppose that \bar{W} denotes the mean value of the measured congestion window size in terms of the number of TCP segments, assuming that the TCP segment size is fixed. In addition, suppose that the congestion window size W at time t is represented by W_t , and the window size in the next RTT is given by W_{t+1} . Then the probability of W_{t+1} being less than or equal to \bar{W}_t satisfies the following inequality:

$$P[W_{t+1} \leq \bar{W}_t] < \frac{8}{3\bar{W}^2 - 2\bar{W}}.$$

Proof: When TCP-Reno⁷ is operating, W changes dynamically. When there is timeout or duplicate ACKs, $W_{t+1} \leq W_t$; otherwise, $W_{t+1} > W_t$ strictly. Now we consider a steady-state TCP connection, whose average measured congestion window size is \bar{W} and whose average threshold is τ when a packet loss event happens. First we consider timeout. In the slow start phase, \bar{W} increases exponentially from 1 to τ , and the length of the slow start phase, a (in RTTs), can be calculated as $a = \log_2 \tau + 1$. In the following collision avoidance phase, whose length is denoted as b (in RTTs), W increases by $\frac{1}{\bar{W}}$ for each RTT from τ to 2τ . Now we calculate a lower bound of b , denoted as b_l . Because $\frac{1}{\bar{W}} \leq \frac{1}{\lfloor \bar{W} \rfloor}$, W increases faster if we assume that W increases by $\frac{1}{\lfloor \bar{W} \rfloor}$ instead of by $\frac{1}{\bar{W}}$ for each RTT. If W increases faster, b becomes less since \bar{W} increases from τ to 2τ during time period b . We calculate the value of b_l under the assumption that W increases by $\frac{1}{\lfloor \bar{W} \rfloor}$ for each RTT. Under this assumption, it takes τ RTTs for W to increase from τ to $\tau+1$. Therefore, for W to increase from τ to 2τ , it takes $b_l = \tau + (\tau+1) + \dots + (2\tau-1) = \frac{3}{2}\tau^2 - \frac{1}{2}\tau$ RTTs. When only timeout is considered, the average window size \bar{W} is less than the maximum congestion window 2τ , i.e., $\tau > \frac{\bar{W}}{2}$. In the time period between two consecutive packet loss events, there is a total of $(a+b)$ RTTs, out of which only one RTT has $W_{t+1} \leq W_t$ and all others have $W_{t+1} > W_t$. Therefore, if we only consider timeout, there is one RTT with $W_{t+1} \leq W_t$ out of $a+b \geq a+b_l = \log_2 \tau + \frac{3}{2}\tau^2 - \frac{1}{2}\tau > \frac{3}{8}\bar{W}^2 - \frac{1}{4}\bar{W} + \log_2 \bar{W} - 1$ RTTs. Next, we consider the case of duplicate ACKs. In this case, the TCP sender directly enters the collision avoidance phase, in which the value of W changes from τ to 2τ , spanning at least $\frac{3}{2}\tau^2 - \frac{1}{2}\tau$ RTTs. Like the timeout case, the relation $\tau > \frac{\bar{W}}{2}$ holds. If we only consider duplicate ACKs, there is one RTT with $W_{t+1} \leq W_t$ out of at least $\frac{3}{2}\tau^2 - \frac{1}{2}\tau > \frac{3}{8}\bar{W}^2 - \frac{1}{4}\bar{W}$ RTTs on average. Finally, if we take both timeout and duplicate ACKs into account, $\min(\frac{3}{8}\bar{W}^2 - \frac{1}{4}\bar{W} + \log_2 \bar{W} - 1, \frac{3}{8}\bar{W}^2 - \frac{1}{4}\bar{W})$ RTTs is a loose lower bound of the time period between two consecutive RTTs in which $W_{t+1} \leq W_t$. In practice, $\log_2 \bar{W} \geq 1$, and therefore a loose upper bound of $P[W_{t+1} \leq \bar{W}_t]$ is given by $\frac{8}{3\bar{W}^2 - 2\bar{W}}$. QED.

⁷The proof is only applicable to TCP-Reno. It is straightforward, however, to generalize the proof to other TCP versions using additive increase multiplicative decrease (AIMD).