Editorial Manager(tm) for Wireless Personal Communications
Manuscript Draft

Manuscript Number: WIRE1457

Title: A Novel Robust Routing Scheme against Rushing Attacks in Wireless Ad Hoc Networks

Article Type: Manuscript

Keywords: Mobile node; On-demand routing protocl; Rushing attack; Wireless ad hoc network.

Corresponding Author: Ms. HyoJin Kim,

Corresponding Author's Institution: Yonsei University

First Author: HyoJin Kim

Order of Authors: HyoJin Kim; Ruy de Oliveira, PhD; Bharat Bhargava, PhD; JooSeok Song, PhD

Abstract: Standard on-demand routing protocols in wireless ad hoc networks were not originally designed to deal with security threats. Because of that, malicious users have been finding ways to attack networks. Rushing attacks represent one of such possibilities. In these attacks, malicious nodes forward the Route Request (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 Medium Access Control (MAC) or routing protocols to get faster processing. As a result, the path through the malicious nodes is chosen, which renders throughput degradation. We propose here a novel, robust routing scheme to defend ad hoc networks against rushing attacks. Our scheme utilizes the "neighbor map mechanism" to establish robust paths as far as rushing attacks are concerned. The proposed scheme also improves path recovery delay by using, whenever it is possible, route maintenance rather than route discovery. Yet, it is energy efficient. The simulation results show that our proposal is indeed viable.

Noname manuscript No.
(will be inserted by the editor)

A Novel Robust Routing Scheme against Rushing Attacks in Wireless Ad Hoc Networks

HyoJin Kim · Ruy de Oliveira · Bharat Bhargava · JooSeok Song

Received: date / Accepted: date

Abstract Standard on-demand routing protocols in wireless ad hoc networks were not originally designed to deal with security threats. Because of that, malicious users have been finding ways to attack networks. Rushing attacks represent one of such possibilities. In these attacks, malicious nodes forward the Route Request (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 Medium Access Control (MAC) or routing protocols to get faster processing. As a result, the path through the malicious nodes is chosen, which renders throughput degradation. We propose here a novel, robust routing scheme to defend ad hoc networks against rushing attacks. Our scheme utilizes the "neighbor map mechanism" to establish robust paths as far as rushing attacks are concerned. The proposed scheme also improves path recovery delay by using, whenever it is possible, route maintenance rather than route discovery. Yet, it is energy efficient. The simulation results show that our proposal is indeed viable.

Keywords Mobile node · On-demand routing protocol · Rushing attack · Wireless ad hoc network

HyoJin Kim
Department of Computer Science, Yonsei University, Seoul, Korea
E-mail: hyojin@emerald.yonsei.ac.kr

Ruy de Oliveira
Department of Computer Science, Federal Institute for Education, Science, and Technology of Mato Grosso (IFMT), Cuiaba, Brazil
Department of Computer Science, Purdue University, West Lafayette, IN 74907, U.S.A.

Bharat Bhargava
Department of Computer Science, Purdue University, West Lafayette, IN 74907, U.S.A.

JooSeok Song
Department of Computer Science, Yonsei University, Seoul, Korea

1 Introduction

In wireless ad hoc networks, a source node transmits packets to the destination node via neighboring mobile nodes (MNs) which have limited power. In these networks, the signals go through bandwidth-constrained wireless links and the routing decisions are determined in a decentralized manner, and so the networks are vulnerable to security threats [1,2].

In on-demand routing protocols, the nodes only forward the first received RREQ packet of a given route discovery procedure [3]. Further RREQ packets belonging to the same route discovery are discarded. This is done to reduce traffic overhead. The problem with this approach lies in its vulnerability to malicious users. Attacks such as "Rushing Attacks" may take advantage of this vulnerability to forward RREQ packets much faster than legitimate nodes [4], leading the routing algorithm to choose the path with the misbehaved nodes. As a result, the nodes may drop packets, which will end up degrading the end-to-end throughput. As detailed in [4], there are many ways for the attackers to get faster processing of the RREQ packets. For instance, they can alter either the MAC or the routing protocols, they can build a tunnel for their path, and so on.

The Rushing Attack Prevention (RAP) [4] approach has been proposed to defend ad hoc networks from rushing attacks. When a neighboring node, of the node running RAP, receives RREQ packets, RAP gathers the received RREQ packets, chooses one of them randomly, and then performs the secure neighbor detection. RAP is energy demanding and complex regarding the secure route delegation between each neighboring node and the destination node.

Another problem with RAP is the high delay in the route discovery phase, in which various RREQ packets are gathered toward randomly selecting an RREQ packet [5][6]. Furthermore, under a path breakage, RAP has to initiate route discovery again instead of simply performing route maintenance. All these features degrade the throughput of RAP.

The authors of [5] evaluated the effects of rushing attacks in ad hoc networks based on the Secured Message Transmission (SMT) [7]. SMT is an end-to-end secure data forwarding mechanism that is executed after route discovery is performed by the Secure Routing Protocol (SRP) [8]. The procedure of [5] to perform SRP and SMT is really complex. In [9], a defense mechanism uses the uncertainty of the intrusion detection. When a node receives an RREQ packet, it signs the packet and forwards it to its neighboring nodes. When a neighboring node receives the RREQ packet, it verifies the signature and then processes it. This scheme requires at least one intrusion detection system for each node, so it is not adequate for wireless ad hoc networks.

This paper proposes a robust routing scheme against rushing attacks aiming to avoid network's throughput degradation. The proposed scheme is adequate for energy-constrained mobile nodes by reducing the number of RREQ packets in the network. Our approach uses the "neighbor map mechanism" in the route discovery to establish robust paths and uses route maintenance to reduce the path recovery delay.

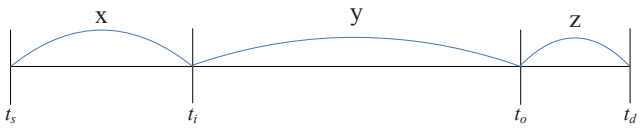

Fig. 1 Relationship between t_s, t_i, t_o, and t_d.

The remainder of the paper is organized as follows: Section 2 shows how rushing attacks can significantly degrade the network throughput. Section 3 presents our proposed scheme. The performance evaluation of our proposal is presented in Sect. 4. Finally, Section 5 concludes the paper.

2 Throughput under Rushing Attacks

Throughput in ad hoc networks can be measured by the packet delivery ratio (PDR), to evaluate the efficiency of our scheme [10]. PDR is computed by the ratio between the amount of transmitted data packets at source node and the amount of such packets received at the destination node. We assume that packets are transmitted during time frame $t_s - t_d$, with generating rate (p) and receiving rate (q).

The actual amount of received packets depends on the topology in place, which may change dynamically, as well as on the robustness of the network against rushing attacks. Rushing attacks will cause problems to real-time traffics such as voice and video that require low jitter to guarantee high Quality of Service (QoS) [11]. By dropping packets, rushing attacks lead the network to experience high jitter.

In general, the routing protocols are optimized to reduce packet transmission delay in the route discovery and route maintenance procedures. The packet transmission delay is critical to PDR in ad hoc networks, and it can be computed using Eq. 1.

$$PDR = \frac{\int_{t_s}^{t_d}(q)dt}{\int_{t_s}^{t_d}(p)dt} = \frac{\int_{t_s}^{t_i}(q)dt + \int_{t_i}^{t_o}(q)dt + \int_{t_o}^{t_d}(q)dt}{\int_{t_s}^{t_d}(p)dt}. \quad (1)$$

Figure 1 shows the relationship among the instants t_s, t_i, t_o, and t_d. The source node starts communicating with the destination node at time t_s and stops at time t_d. At time t_i, route maintenance is started or route discovery is restarted. At time t_o, the path is repaired completely without any path breakages and attacks. In Fig. 1, the interval x, which refers to the *normal-period*, starts when a path is initially established at t_s by route discovery and ends at t_i when the data transmission starts before either the path is broken or a rushing attacks is conducted. The interval y corresponds to the sum of all *detection-recovery* intervals, which include the time for detecting the broken path, re-establishing the path, and resuming data transmission.

The interval z represents the *stable-period*, in which the data transmission

is started and stopped without interruptions. This means that the path is not broken any longer. The interval z may not even exist if the path is not repaired until the time t_d.

By associating the intervals of Fig. 1 with the *PDR*, it is proper to say that *PDR* depends on the length of the interval y. The longer the interval y is, the lower the *PDR*. Hence, in order to keep *PDR* high, the interval y has to be reduced. This can be accomplished by using route maintenance, after a path interruption, instead of route discovery again, and by establishing robust paths against rushing attacks. And this is exactly what our scheme is proposed for.

3 Novel Robust Routing Scheme against Rushing Attacks in Wireless Ad Hoc Networks

We assume that all wireless links are bidirectional, so a source node can send an RREQ packet and receive its RREP packet back using the same path. We also assume that there is always at least one legitimate node around any legitimate node. This is crucial to avoid that multiple adjacent misbehaved nodes maliciously report to the source node that the legitimate node is hazardous.

Generally, the shortest path is defined as a path that has the least number of hops, so the RREP packet ($RREP_1$) going through such a path arrives at the source node faster than the RREP packets coming from other paths. We denote $PATH_1$, $PATH_2$,$\cdots$, $PATH_j$ as paths whose RREP packets arrive at the source node in order. When a rushing attack is conducted, the source node may receive an RREP packet ($RREP_M$) sent by the malicious node faster than other RREP packets.

Our approach uses three special kinds of packets to protect the network from rushing attacks as follows:

- Route Check (**RCHE**) packet - for measuring the round trip time (RTT) of the current path;
- Route Verify (**RVER**) packet - for verifying whether the measured RTT is proper to use or not;
- Route Failure (**NFAI**) packet - for notifying the source node that a node has failed.

Figure 2 illustrates a possible scenario for assisting us on the explanations that follow. In Fig. 2, the source node N_S wants to send data to the destination node N_D. N_{M1}, N_{M2}, and N_{M3} are malicious nodes. N_{M1} and N_{M3} are neighbors of N_{M2}. They are positioned in a path N_S-N_{M1}-N_{M2}-N_{M3}-N_D that allows them to conduct rushing attacks by forwarding RREQ packets faster than other paths. They do so by hiding the existence of the malicious node N_{M2}. In fact, they report the route N_S-N_{M1}-N_{M3}-N_D to the destination node.

In this situation, despite the fact that the path N_S-N_B-N_E-N_F-N_D is actually the shortest path, in reality the path N_S-N_{M1}-N_{M2}-N_{M3}-N_D is the one chosen as the path to transmit data. When node N_{M2} receives data packets,

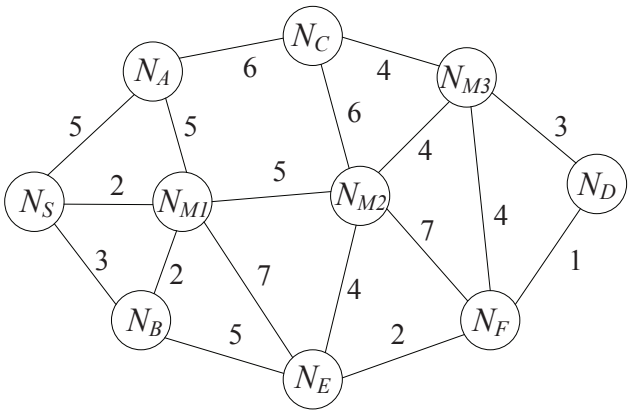

Fig. 2 An example topology: the topology is established from N_S to N_D. Each number between adjacent nodes is the relative distance measured by Eq.3.

it may drop them and not forward them to N_D.

In our proposed scheme, the source node uses two thresholds, α and β, as presented in Eq. 2, to choose robust paths in route discovery and route maintenance phases, as it is further explained later.

$$\begin{aligned} \alpha &= 2 \times \min\{T(RREP_1), T(RREP_2), \cdots, T(RREP_k)\}, \\ \beta &= \min\{T(RREP_1), T(RREP_2), \cdots, T(RREP_k)\} \\ &\quad + \text{mean}\{T(RREP_1), T(RREP_2), \cdots, T(RREP_k)\} \end{aligned} \tag{2}$$

where k is the number of received RREP packets at the source node and T($RREP_p$) is the received time of the $RREP_p$ packet at the source node.

Our scheme also uses the "neighbor map mechanism." In this mechanism, the source node has the *neighbor map* which is a table containing the map information of the nodes in the network. Each row of the table is associated to a distinct node. Hence, the number of rows equals the number of node minus 1. The neighbor map mechanism is built by the nodes sending their neighbors information to the source node, and such information are, for each node, composed of: $<$ *node id, nodal arms (neighbors) of the node, relative distance of each neighbor to the node, updated times of the relative distances* $>$.

In the neighbor map, the "nodal arms" of a given node is defined as the neighboring nodes, which are one-hop away, from such a node. For example, in Fig. 2, nodes N_S, N_A, N_B, N_E, and N_{M2} are nodal arms of N_{M1}.

The relative distance is associated to the nodes' neighbors. It represents the elapsed time since a RREQ packet was transmitted by the neighboring node N_l up to the time when the node N_l' received it. For each node, the relative distance $D_{l-l'}$ is calculated as shown in Eq.3.

$$D_{l-l'} = |T_R(RREQ < N_l - N_{l'} >) - T_T(RREQ < N_l - N_{l'} >)| \quad (3)$$

where $T_T(RREQ < N_l - N_{l'} >)$ is the time when node N_l transmits the RREQ packet to node $N_{l'}$ and $T_R(RREQ < N_l - N_{l'} >)$ is the time when node $N_{l'}$ receives the RREQ packet.

Then, we measure the relative distance of a path (*D(PATH)*) by summing all relative distances of adjacent nodes as:

$$D(PATH) = \sum_{p=1}^{q-1} (D_{p-p'}). \quad (4)$$

We denote q as the number of nodes on the path. *D(PATH)* is compared to the thresholds α and β in order to establish a robust path against rushing attacks as shown later.

3.1 Route Discovery

In the route discovery, the following steps are to be executed.

1. When N_S, in Fig. 2, wants to transmit data to N_D, it generates RREQ packets marked with an R flag, to indicate the use of our proposed scheme and forwards the packet to its neighbors so as to establish a path.
2. When a given node receives an RREQ packet with the R flag, the node processes only the first received RREQ packet for that path. Then, the node randomly processes all other packets it receives, so receivers do not need to wait for a random time. This leads the route discovery delay to be lower than in other schemes such as RAP.
3. Each node stores the RREQ packet received time, the packet ID, the source node ID, the list of nodal arms, and the RREQ packet transmission time. For example, the stored information of node N_{M1} in Fig. 2 is $T_R(RREQ < N_S - N_{M1} >)$, $T_T(RREQ < N_{M1} - N_A >)$, $T_T(RREQ < N_{M1} - N_B >)$, $T_T(RREQ < N_{M1} - N_E >)$, $T_T(RREQ < N_{M1} - N_{M2} >)$, RREQ ID, N_{M1} ID, and the nodal arms (N_S, N_A, N_B, N_E, N_{M2}). Each node receives an RREQ packet, writes the current transmission time and its list of nodal arms in the RREQ header and forwards it to the neighbors.
4. When the destination node receives the RREQ packet, it generates an RREP packet containing its nodal arms and the RREQ received and transmitted time and then sends the packet back to the source node.
5. If the source node receives two or more RREP packets with different path information such as $PATH_1$ and $PATH_2$, it draws the neighbor map. As an example, the following tuples show the content of the neighbor map for N_{M1} in Fig. 2: < N_{M1}, N_S, 2, 10>, < N_{M1}, N_A, 5, 20>, < N_{M1}, N_B, 2, 18>, < N_{M1}, N_E, 7, 15>, < N_{M1}, N_{M2}, 5, 22>. With this neighbor

map, the source node has the virtual topology of the network. The more information the source node gets, the higher the accuracy of the neighbor map.

6. In order to protect the system from rushing attacks, the source node chooses only reliable paths. Moreover, the source node chooses the path with the minimum number of hops among the chosen reliable paths. Then, it chooses a path whose *D(PATH)* is less than α and the average number of nodal arms is larger than three. The former guarantees that the selected path is really a short one, and the latter assures a robust path, as for the nodes with more than three nodal arms it is possible, in case of a broken link, to forward packets to alternate links. If the source node cannot find a path, it has to wait for a random time to collect other path information.
7. The source node transmits data onto the chosen path.
8. The source node randomly sends an RCHE packet to check the RTT of the path.
9. When a given node receives the RCHE packet, it writes the received and transmitted time and forwards the packet to the next node.
10. When the destination node receives the RCHE packet, it generates an RVER packet, containing the updated received and transmitted time of all nodes on the path, back to the source node.
11. The source node updates the neighbor map with the information in the RVER packet.

The source node monitors the path using RCHE and RVER packets until the data transmission is completed.

3.2 Route Maintenance

When a path is broken during data transfer, it has to be repaired. As shown in Fig. 3.2, a failed node (N_H) may send a NFAI packet to the source node (N_G) via the previous node (N_{H-1}). Alternatively, as shown in Fig. 3.2, node N_{H-1} can create an NFAI packet if node N_H is not found in its radio transmission range.

If the node N_{H-1} has more than three nodal arms, it may forward data to all nodal arms except to node N_H. When the source node receives the NFAI packet, it chooses another path whose nodal arms are more than three and *D(PATH)* is less than β. This assures a short and robust path.

By using this algorithm, the packets are sent onto the new path, reducing the number of flooding control packets in comparison with what happens if the route discovery procedure is invoked again. If there is no path satisfying the requirements, the source node performs route discovery once again, as done in schemes as such as RAP. After that, the source node updates α through the value of β.

Route maintenance is not only needed under path breakage, but also needed under a rushing attack. Our proposed scheme avoids rushing attacks by using the neighbor map mechanism. This mechanism chooses only reliable paths in

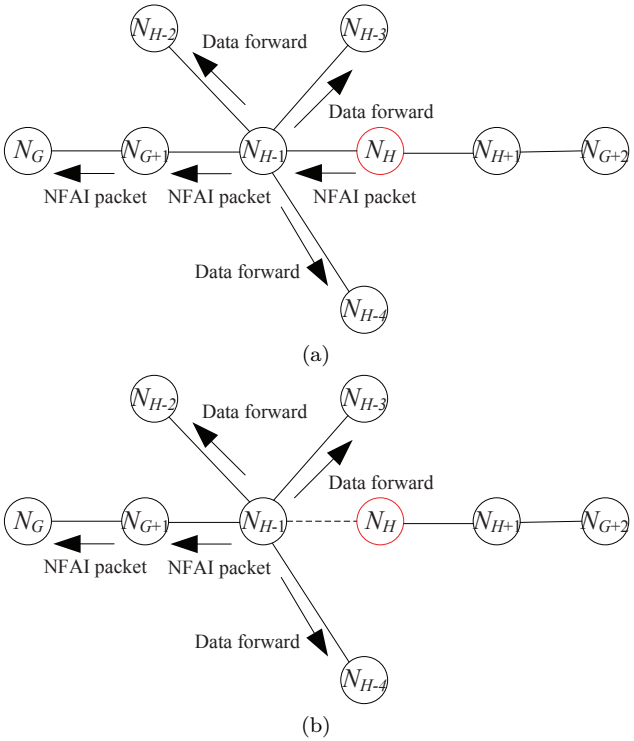

Fig. 3 Examples of a node failure: (a) NFAI packet is sent by the failed node. (b) NFAI packet is sent by the previous node of the failed node.

route discovery by measuring the RTT of the paths using RCHE and RVER packets to avoid rushing attacks.

Besides, the source node always chooses a new path with $D(PATH) < \beta$ whenever either the RTT$> \alpha$ or the packet transmission delay between adjacent nodes $Delay_{N(j)\text{-}N(j+1)} > RTT_{limit}$ which is presented in Eq.5.

$$RTT_{limit} = 2 \times A(T_{T<N_{(j)}-N_{(j+1)}>}(packet)) \tag{5}$$

where $A(T_{T<N_{(j)}-N_{(j+1)}>}(packet))$ is the average packet transmission delay from the node N_j to its next node N_{j+1}.

After performing the route maintenance, α and β are updated.

4 Evaluations

We evaluated the performance of the proposed scheme, comparing it with DSR and RAP [4]. We used the ns-2 simulator [12] in all the experiments. We simulated the DSR routing protocol, which is a basic on-demand routing protocol that has no specific security mechanisms. We also simulated the RAP

protocol because it is a well-known defense protocol for rushing attacks. In fact, we used RAP 1 Flow [4] for simplicity. Throughout this paper, we use the term RAP instead of RAP 1 Flow, though.

In these experiments, we used 100 nodes which randomly move according to the random waypoint model [13] within a 1000 m $\times$ 1000 m area for 900 seconds. Each node starts placed randomly within the area and then waits sequentially for a pause time of 0, 30, 60, 120, 300, 600, and 900 seconds. After waiting the pause time, the node uniformly chooses its velocity between 0 and 20 meters per second. The node moves with the chosen velocity and then waits for the next pause time again. The node repeatedly moves and waits until the simulation is finished.

For the traffic generator, we used source as a Constant Bit Rate (CBR) flow at 2 Mbps over the UDP protocol. Among the 100 nodes, we chose 20 nodes for source nodes and 20 nodes for destination nodes. Then, 40 nodes were chosen as legitimate nodes and 20 nodes as rushing attackers. The attackers were designed based on [7], so they do not forward anything except routing related packets.

4.1 Transmission Delay

Figure 4 shows the transmission latency varying with pause time. We define the transmission latency as a delay between a path request time and the data transmission completion time. The results are as expected, since under high mobility speed (low pause time), the topology changes quickly due to the continuing movement of the nodes. These changes demand path re-establishment, which explains why all schemes experienced mostly longer transmission delay as pause delay increased. As shown in the Fig. 4, the proposed scheme has the best performance. This happens because it only waits for the second RREP packet before initiating route discovery and performs route maintenance rather than route discovery.

The worse performance for the other two protocols may be explained as follows. Both RAP and DSR have longer transmission delay because they initiate route discovery instead of route maintenance whenever the topology is changed. RAP spends significant amount of time collecting RREQ packets to perform the secure neighbor detection, which causes the interval y, in Fig. 1, to be higher than the one in our scheme. The highest transmission delay for DSR is due to its lack of specific secure mechanisms to deal with rushing attacks. Because of that, its interval y gets really high in comparison with the other two evaluated schemes.

4.2 Energy Consumption

The energy of a mobile node is consumed by transmitting, receiving, or processing packets. We focus here on energy consumed for transmitting packets

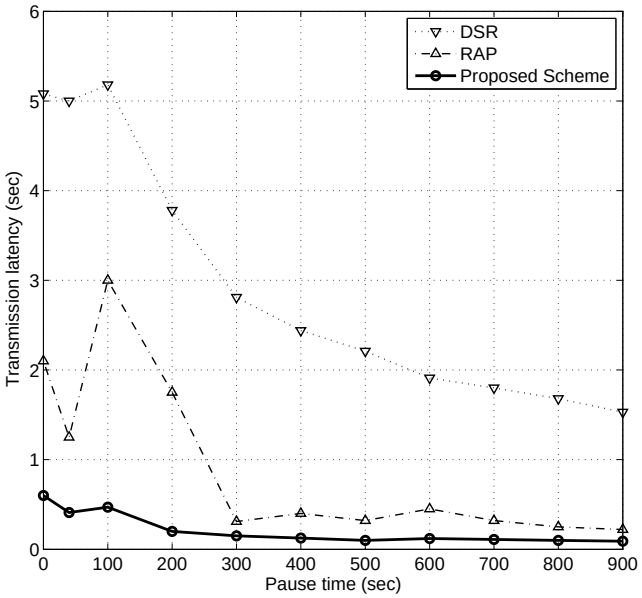

Fig. 4 Transmission latency vs. pause time.

because the other two consumptions are much smaller than it [14]. Equation 6 can be used to compute the transmitting power P_t of a node with receiving power P_r. In fact, each node can receive signal within a radius (d) and generally has a path loss exponent (a) between 2 and 4, depending on the characteristics of the communication medium [15].

$$P_r = P_t \times d^{-a}. \tag{6}$$

For a certain period of time, a node consumes the power P_t to transmit both data and control packets. Nevertheless, as we want to underline the low energy, or power, consumption of our scheme, we address only the control packets in the discussions below. Hence, the amount of power consumption of a node to transmit only control packets (PW) is given by Eq. 7.

$$PW = PW_{RREQ} + PW_{RREP} + PW_{RERR} + PW_{NFAI} + PW_{RCHE} + PW_{RVER} \tag{7}$$

where PW_{RREQ}, PW_{RREP}, PW_{RERR}, PW_{NFAI}, PW_{RCHE}, and PW_{RVER} are the amount of power to transmit RREQ, RREP, RERR, NFAI, RCHE, and RVER packets, respectively.

Even though we have not measured the power consumption in the simulations, Eq. 7 allows us to infer important observations. By this Equation it is clear that the transmitting power depends on the number of packets transmitted. Yet, the NFAI, RCHE, and RVER packets are much smaller than the RREQ, RREP, and RERR packets. This confirms that our scheme is energy

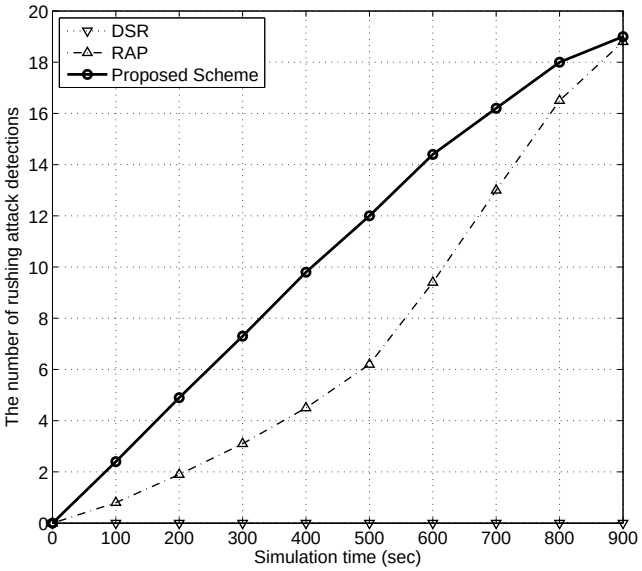

Fig. 5 The number of rushing attack detections vs. simulation time.

saving since it reduces the number of the three bigger packets. It does so by forcing route maintenance instead of route discovery. Its robust paths imply less RERR packets as well. All these details permit us to affirm that our scheme outperforms both DSR and RAP in terms of energy consumption.

4.3 Number of Rushing Attack Detections

The experiment addressed in this subsection was carried out to see how effective our scheme was in detecting launched attacks. Fig. 5 presents the number of rushing attack detections varying with simulation time.

Note that DSR cannot detect any attack. RAP detects the attacks but is less effective than our scheme, especially at the beginning of the simulation. This happens because RAP spends non-negligible amount of time collecting RREQ packets. Overall, the mechanism of our proposed scheme rendered it very efficient here as well. In fact, the interval y of our scheme is shorter than the one of the others because it establishes robust paths with the nodal arms and monitors the paths using RCHE and RVER packets.

4.4 Packet Delivery Ratio

Figure 6 shows the packet delivery ratio varying with pause time. In the lower pause time, the three schemes have the lowest packet delivery ratio because the nodes move dynamically. Again our proposed scheme achieved the best performance. This occurs because as far as the interval y is concerned that

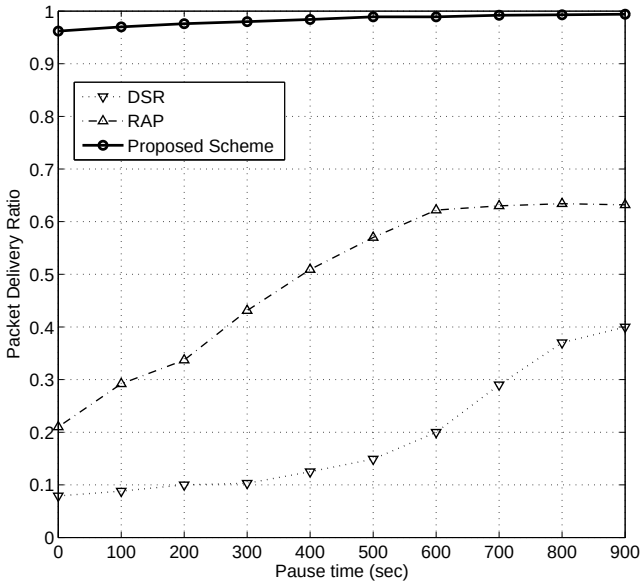

Fig. 6 Packet delivery ratio vs. pause time.

it has the shortest length in our scheme, which is a result of the fast route maintenance algorithm in it. Yet, the high packet delivery ratio of our scheme is possible due to the reduced number of path breakages that it provides the robust paths.

5 Conclusion

We have proposed a defense mechanism against rushing attacks in ad hoc networks. Our scheme establishes robust paths by using the "neighbor map mechanism" which provides the source nodes with enough information about the routes to both detect and avoid rushing attacks. The proposed scheme speeds up link recovery as it prioritizes path maintenance over path recovery. The evaluation results show that our scheme provided outstanding performance in terms of transmission latency, amount of simultaneous attackers, and packet delivery ratio. Besides, our mechanism outperformed the compared existing work in all evaluated situations. It was also shown through analysis that the proposed scheme is energy efficient, which is fundamental for the presumably battery-powered devices in place. For future work, it would be interesting to implement and evaluate our algorithm in a real-life network. Scenarios with higher number of mobile nodes should also be examined. A quantitative evaluation of the real energy consumption benefits of our proposed scheme could be conducted as well.

References

1. Xia L., Slay J. (2008). Securing Wireless Ad Hoc Networks: Towards A Mobile Agent Security Architecture.
2. Deng H., Li W., Agrawal D. P. (2002). Routing security in wireless ad hoc networks. *IEEE Communications Magazine*, 40(10), 70–75.
3. Agarwal K., Awasthi L. K. (2008). Enhanced AODV Routing Protocol for Ad hoc Networks. *16th IEEE International Conference on Networks*, 1–5.
4. Hu Y.-C., Perrig A., Johnson D. B. (2003). Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocol. *2nd ACM workshop on Wireless security*, 30–40.
5. Rawat A., Vyavahare P. D., Ramani A. K. (2005). Evaluation of Rushing Attack on Secured Message Transmission (SMT/SRP) protocol for Mobile Ad hoc Networks. *2005 IEEE International Conference on Personal Wireless Communications*, 62–66.
6. Kent S., Lynn C., Seo K. (2000). Secure Border Gateway Protocol (S-BGP). *IEEE Journal on Selected Areas in Communications*, 18(4), 582–592.
7. Papadimitratos P., Haas Z. J. (2003). Secure message transmission in mobile ad hoc networks. *Ad Hoc Networks 1*, 193–209.
8. Papadimitratos P., Haas Z. J. (2002). Secure Routing for Mobile Ad hoc Networks. *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference*, pp.193–204.
9. Dreef D., Ahari S., Wu K., King V. (2004). Utilizing the Uncertainty of Intrusion Detection to Strengthen Security for Ad Hoc Networks. *3rd International Conference on AD-HOC Networks and Wireless*, LNCS 3158, 82–95 .
10. Thriveni J., Akekhya V. L., Deepa N., Uma B., Alice A., Prakash G. L., Venugopal K. R., Patnaik L. M. (2008). QoS Preemptive Routing with Bandwidth Estimation for Improved Performance in Ad Hoc Networks. *4th International Conference on Information and Automation for Sustainability*, 443–448 .
11. Nakamura S., Ohta K., Kato N. (2001). Proposal of Dynamic Bandwidth Allocation Technique for Low Delay/Low Jitter Realtime Communication and Its Evaluation by Using CBQ, *IEICE transactions on communications*, E84-B(6), 1513–1520.
12. The VINT Project: The network simulator – ns-2. Available at http:// www.isi.edu/nsnam/ns/.
13. Johnson D. B. (1994). Routing in Ad Hoc Networks of Mobile Hosts. *IEEE Computer Society Workshop on Mobile Computing Systems and Applications*, 158–163.
14. Kim H., Han S., Song J. (2006). A Routing Protocol for Throughput Enhancement and Energy Saving in Mobile Ad Hoc Networks. *International Conference on Computational Science and Applications*, LNCS 3981(2), 359–368.
15. Cheng M. X., Cardei M., Sun J., Cheng X., Wang L., Xu Y., Du D.-Z. (2004). Topology Control of Ad Hoc Wireless Networks for Energy Efficiency. *IEEE Transactions on Computers*, 53(12), 1629–1635.

Your PDF file "rushing.pdf" cannot be opened and processed. Please see the common list of problems, and suggested resolutions below.

Reason:

Other Common Problems When Creating a PDF from a PDF file

You will need to convert your PDF file to another format or fix the current PDF file, then re-submit it.

***author's picture & biography**

Click here to download author's picture & biography: authors.doc

Hyojin Kim received the B.S degree and the M.S. degree in Computer Science from Yonsei University, Seoul, Korea, in 2002 and 2004. She is currently a candidate of Ph.D. degree in Computer Science at Yonsei University, Seoul. Korea. Her research interests include information security and ad hoc networks.

Ruy de Oliveira graduated from the Federal Engineering School of Itajuba (EFEI), Brazil, as an Electro nic Engineer, in 1992. In early 1999 he joined the Computer Networks Laboratory team at Electrical En gineering Faculty of the Federal University of Uberlandia (UFU), where he received his MS degree in c omputer networks area in February 2001. From April 2001 to June 2005 he worked as a research assist ant at the Computer Networks and Distributed Systems group (RVS) under the Institute of Computer Science and Applie d Mathematics (IAM) of the University of Berne (Unibe), where he was involved in some projects and pursued his PhD degree. He is currently joined with Department of Computer Science, Federal Institute for Education, Science, and Technology of Mato Grosso (IFMT), Cuiaba, Brazil and Department of Computer Science, Purdue University, West Lafayette, IN 74907, U.S.A.

Bharat Bhargava is a professor of the department of computer science and department of electrical & c omputer engineering at Purdue university since 1984. He is conducting research in security issues in mo bile and ad hoc networks. He serves on five editorial boards of international journals. He was the chair man of the IEEE Symposium on Reliable and Distributed Systems held at Purdue in October 1998. He is a Fellow of the Institute of Electrical and Electronics Engineers and of the Institute of Electronics and Telecommunicat ion Engineers. He has been awarded the charter Gold Core Member distinction by the IEEE Computer Society for his di stinguished service. He received Outstanding Instructor Awards from the Purdue chapter of the ACM in 1996 and 1998. In 1999, he received IEEE Technical Achievement award for a major impact of his decade long contributions to foundati ons of adaptability in communication and distributed systems. In 2003, he has been inducted in the Purdue's book of gre at teachers.

JooSeok Song received the B.S degree in Electrical Engineering from Seoul National University, Seoul, Korea, in 1976, and the M.S. degree in Electrical Engineering from KAIST, Korea, in 1979. In 1988, he received the Ph.D. degree in Computer Science from University of California at Berkeley. He is currently a Professor of Computer Science at Yonsei University, Seoul. Korea. His research interests include information security and wireless communication.