Reliable and Secure Congestion Aware Routing Protocol for Wireless Sensor Network

Neelufar S[1], Sangeetha D S[2]

Department of Computer Science, Sri Bhagawan Mahaveer Jain College, Karnataka, India

[1]Email: *neelufarshabana[at]gmail.com*
[2]Email: *sangeethavenkatesh1[at]gmail.com*

Abstract: ***Mobile adhoc networks sense various kinds of information, process them locally and communicate it to the outside world via Internet. In the near future, sensor networks will play a major role in collecting and disseminating information from the fields where ordinary networks are unreachable for various environmental and strategically reasons. Hence it is increasingly likely that sensors will be shared by multiple applications and gather heterogeneous data of different priorities. With such concentration on mobile adhoc networks, vital issues like security and congestion control are to be taken care of. We propose Secure Congestion Aware Routing protocol (RSCARP), a protocol designed for resolving congestion by dedicating a portion of network to forward high-priority traffic primarily and also satisfies the major security properties like data authentication, data secrecy, replay protection, freshness with low energy consumption which are the major factors affecting the mobile adhoc networks.***

Keywords: Congestion Aware Routing, Mobile adhoc networks and routing optimization

1. Introduction

There is an electrifying new wave in sensor applications wireless sensor networking which enables numerous sensors and actuators to be deployed independent of the costs and physical constraints of wiring, opening up a new world of sensing application possibilities. The ad hoc nature of wireless mesh networks enables the sensor nodes form a network automatically with minimal human interference. However, energy possessed by sensor nodes is limited, which becomes the most challenging issue in designing sensor networks [3]. The main power consumptions in sensor networks are computation and communication between sensor nodes . In particular, the ratio of energy consumption for communication and computation is typically in the range of 400. Therefore it is critical to enable mutual information processing and data aggregation to prolong the lifetime of sensor networks. Minimizing the communication costs between sensor nodes is critical to lengthen the lifetime of sensor networks. In other words, we should carefully select sensor nodes to contribute in the task.

2. Related Work

An obvious solution to enhance service to HP data is to use priority queues to provide differentiated services However, in such schemes, though HP packets get precedence over LP packets within a node, at the MAC layer, they still compete for a shared channel with LP traffic sent by surrounding nodes. As a result, without a routing scheme to address the impact of congestion and hotspots in the network, local solutions like priority queuing are not sufficient to provide adequate priority service to important data. QoS in sensor networks has been the focus of current research.

Degrading service to one type of data to provide better service to another has been used in schemes like RAP [2] and SWAN Many of the schemes do not adopt differentiated routing, which leverages the large uncongested parts of the network that is often underutilized to deliver LP traffic. Hence we use differentiated routing to provide the best possible service to HP data while trying to decrease the energy consumption in the conzone.

Congestion in sensor networks has been addressed in works like CODA and Fusion.

Though these schemes take important steps to mitigate congestion in sensor networks, they treat all data equally. Priority based schemes have been addressed in CAR and MCAR.

Several encryption modes exist that achieve secrecy and authentication. We select OCB as our encryption mode since it is especially well-suited for the stringent energy constraints of sensor nodes. In addition to OCB, RSCARP also uses loose time synchronization to minimize energy consumption and can be used to provide efficient replay protection in broadcast communication similar to Minisec. In this section, we briefly review OCB.

OCB, or Offset Code Book, is a block-cipher mode of operation that features authenticated encryption. Given a plain text of arbitrary length, OCB generates a cipher text that simultaneously provides authenticity and data secrecy. OCB is provably secure, and is parameterized on a block cipher of block size *n* and a tag of length t. t is defined such that an adversary is able to forge a valid cipher text with probability of 2^{-t}. OCB operates as follows. OCB is especially well suited for sensor nodes. OCB avoids cipher text expansion. OCB has superior performance, since it provides secrecy and authenticity in one pass of the block cipher. TinySec and ZigBee provide the same security guarantees, but require two passes of the block cipher: one pass achieves secrecy with CBC-encryption, and another pass achieves authenticity with CBC-MAC. Consequently, since TinySec almost doubles the amount of computation, the energy consumption also doubles. OCB requires very few block cipher calls when compared to CBC-encryption and CBC-MAC schemes.

3. RSCARP: Reliable and Secure Congestion Aware Routing Protocol

We present RSCARP, a secure congestion aware routing protocol that discovers the congested zone of the network that exists between high-priority data originators and the data consumers. Using simple forwarding rules, this portion of the network is dedicated to forward high-priority traffic primarily. It also satisfies all the security properties like data authentication, data secrecy, replay protection, freshness with low energy consumption which are the major factors affecting the Mobile adhoc networks.

3.1 Overview

An important event occurs in one portion of the sensor field called the critical area. This critical area will typically consist of multiple nodes. In such a scenario, there is a data processing center for collecting sensitive information from the critical area called sink. We refer to the area that contains the shortest paths from the critical area to the sink as the conzone. Our basic solution, called RSCARP, operates solely in the network layer. Packets are classified as HP or LP by the data sources, and nodes within a conzone only forward HP traffic. LP traffic is routed out of and/or around the conzone. Also, the sources employ OCB-encryption to encrypt the data before communicating the data in order to provide security. The sink, upon receiving the packet decrypts thedata.

3.2 Description

In this section, we describe the mechanisms of RSCARP in the network scenario where there are multiple sources and a single high priority sink.

RSCARP comprises of the following five steps:

1) Formation of High-Priority Routing Network

After the deployment of sensor nodes, Sink initiates the process of building the HP routing network (HiNet). This network covers all nodes, because at the time of deployment, the sink will usually have no information on the whereabouts of the critical area nodes. Since all HP data is destined to a single sink, the HiNet is based on a minimum distance spanning tree rooted at the sink. A node that has multiple neighbors with depths (the number of hops to the sink) less than its own considers them all as parents.

We now consider the HiNet formation process. Once the sink discovers its neighbors, it broadcasts a "Build HiNet" message (containing the ID and depth of the node) asking all nodes in the network to organize as a graph. Once a neighboring node hears this message, it checks if it has already joined the HiNet (i.e., if it knows its depth); if not, it sets its depth to one plus the depth in the message received and sets the source of the message as a parent. Similarly this node then rebroadcasts the Build HiNet message, with its own ID and depth. If a node is already a member of the graph, it checks the depth in the message, and if that depth is one less than its own, then the source of the message is added as a parent. In this case, the message is not rebroadcast. Finally, the Build HiNet message is rebroadcast with the new depth value. In this fashion, the Build HiNet message is sent down the network until all nodes become part of thegraph.

2) Conzone Discovery

Nodes discover if they are on the conzone by using the conzone discovery mechanism. This conzone discovery is done dynamically, because the critical area can change during the lifetime of the deployment and is triggered when an area starts generating HP data. A conzone must be then discovered from that neighborhood to the sink for the delivery of HP data. To do this, critical area nodes broadcast "discover conzone to sink" (ToSink) messages. This message includes the ID of the source and its depth and is overheard by all neighbors. When a node hears more than α_s (neighborhood size) distinct ToSink messages coming from its children, it marks itself as on conzone and propagates a single ToSink message. For node x with depth d_x and neighborhood size n_x, setting correctly for different depths ensures that the conzone is of an appropriate width.

ToSink Threshold : α_s = þdx . d_s . n_s

An important goal of the conzone discovery algorithm isto split the parents and siblings (nodes with the same depth) in the HiNet into on-conzone and off-conzone neighbors. Since the presence of a conzone leads to suboptimal routing for LP data due to on-conzone nodes being dedicated to serving HP data, after the HP stream comes to an end, the conzone is destroyed by flooding a "destroy conzone" message in the conzone.

3) Encryption

We assume that symmetric keys KÆiB , KBÆi are already established between each source A_i and the sink B. We recommend a different key for each source, but our protocol is by no means restricted to such a setup. A monotonically increasing counter is assigned to each key as the Initialization Vector IV (CÆiBused to for key KÆiB), and is kept as internal state by both sender and receiver.

We employ OCB-encryption with the packet payload as M, packet header as H, counter CÆiB as the nonce, and KÆiB as the encryption key. We selected Skipjack to be the underlying block cipher with a block size of 64 bits. Since OCB requires the nonce to be the same length as the block size, counter CÆiB can also be 64 bits. Alternatively, the counter could be of shorter length, and be padded out to 64-bits when requested by the OCB encryption function. The second parameter of OCB is the tag length τ, which we set to 32 bits, a length suitable for security in retail banking . Sink B maintains a buffer of counters CÆ1B, CÆ2B... CÆiB for all the sources A_1, A_2,...A_n. The source increments its counter value by one before sending each message.

4) Differentiated Routing

After the message is encrypted, HP data is routed in the conzone and LP data is routed off the conzone. LP data generated inside the conzone is routed out using the following approach. When an on-conzone node gets an LP message, it forwards it to an off-conzone parent, if there are any. Otherwise, the LP data is forwarded to an off-conzone sibling. If there are no parents or siblings that are off conzone, we resort to the following method. After discovering the conzone, the sink sends a message through the conzone, which contains the coordinates of a line that cuts the conzone in half. This line connects the sink to the center of the critical area. Using this information and its own coordinates, a node can determine on which half of the conzone it lies and hence routes LP data to the parent that is closest to the conzone boundary, i.e., farthest from the line. With the assumption of uniform deployment density, this ensures that all LP data generated inside the conzone is routed out efficiently and along the shortest path.

We used AODV in the off-conzone nodes to route LP data, with the modification that the on-conzone nodes do not propagate route request or reply messages for LP data. Using this modified routing scheme, LP data generated outside or routed out of the conzone is routed to its destination via off-conzone nodes only.

5) Decryption

Upon receiving the message, sink B decrypts it with the key KBÆi of the corresponding sender A_i. Sink then increments its local copy of CÆiB accordingly so that it remains consistent with A_i.

3.3 Security analysis

In this section, we provide an analysis on the level of security promised by RSCARP.

1) *Secrecy:* Semantic security requires that nonces do not repeat. In RSCARP, the counter is kept as internal state, and thus can be made arbitrarily long. We choose 8 bytes, which means that the nonce would not repeat until after sending 2^{64} messages.
2) *Replay protection:* Each sender and receiver keeps a synchronized counter that is used as the nonce in OCB encryption. The receiver would only accept messages with higher counter values than the those maintained in the node state. Thus, replayed packets will all be rejected.
3) *Freshness:* In RSCARP, the receiver can arrive at the counter value used for each packet by verifying the validity of OCB decryption. The receiver can use the counter value of two messages to enforce message ordering, thus providing freshness.

Algorithm

Local Variables

Off-conzone parents: P_{off} = {p_1, p_2,………, p_n} Off-conzone siblings: S_{off} = {s_1, s_2,………, s_n} On-conzone parents: P_{on} = {}
On-conzone siblings: S_{on} = {} Children: Children = {c_1, c_2,………, c_k}

Node's on-conzone status: On_conzone = FALSE ToSink message received: ToSink_received 0 ToSink threshold: α_s = þd_x . d_s . n_s

Conzone Discovery :

```
if node x receives ToSink from child c1 then if On_conzone == FALSE then
if ToSink_received > αx then
On_Conzone = TRUE
if x is not sink then broadcast ToSink with dx
else
ToSink_received ++
else if node x receives ToSink from parent pj then
Poff - = {pj}; Pon + = {pj}
else if node x receives ToSink from sibling si then
Soff - = {si}; Son + = {si}
```

Encryption:

```
//Encrypt the payload M with the symmetric key KÆiB
Encrypt(M, KÆiB) Counter CÆiB ++
Assign the counter as nonce to key as IV
if length of CÆiB < 64 bits then
Pad the counter with zeros
//Set tag length τ τ = 32 bits
```

Differentiated Routing: if P_{on} ≠ {} then

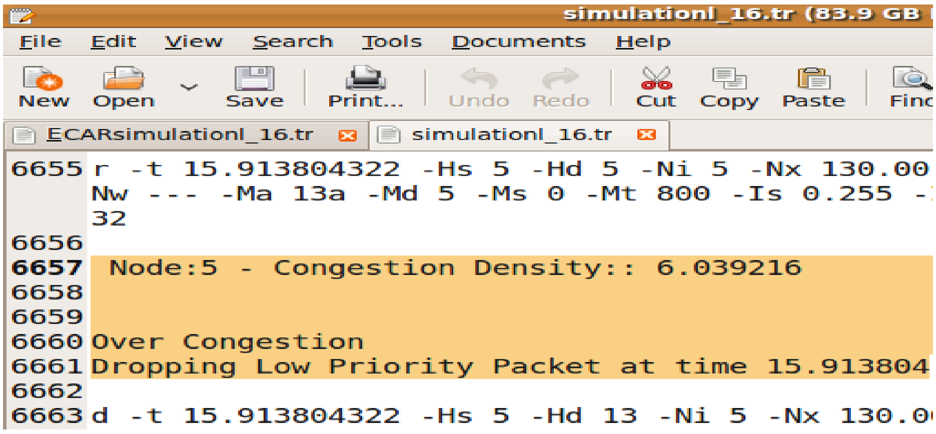
```
Send data to any p є Pon
else if a sibling s є Son then send data to s
else
send data to any u є Poff U Soff
```

Decryption:

```
//Decrypt the message with the
key KBAi Decrypt(M, KBÆi)
Identify the counter of the source Ai
//Increment the local copy of CÆiB Counter CÆiB++
```

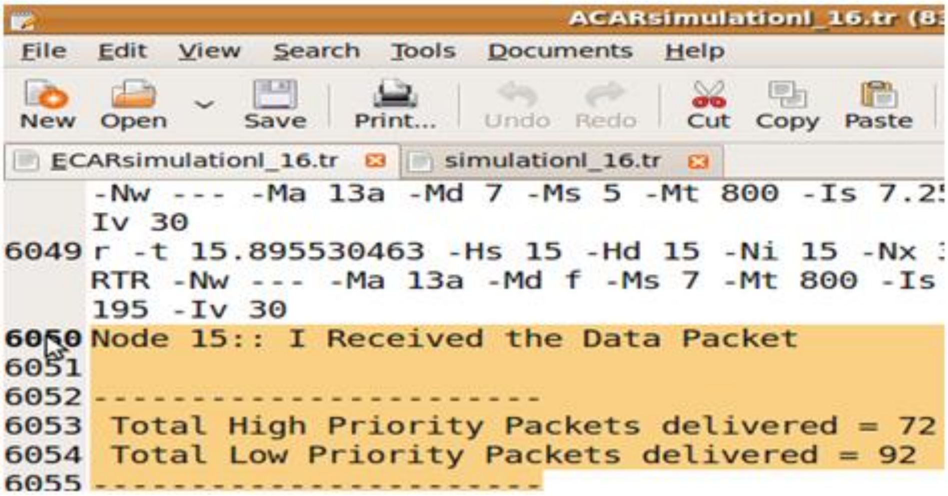
4. Experimental Results

For the simulation, we create a square flat platform of finite dimensions for simulation. Various parameters are kept permanent while others are varied to help us analyze the performance of the three protocols. The simulation is done in the random waypoint model in a rectangular field. The field configurations used is: 400 m x 400 m field with 8, 111, 21, 811 and 48 nodes. Here, each packet starts its journey from a random location to a random destination with a randomly chosen speed (uniformly distributed between 0–20 m/s). Once the destination is reached, another random destination is targeted after a pause. The pause time, which affects the relative speeds of the mobiles, is varied. Simulations are run for 14 simulated seconds. We do the simulation work with taking different no. of nodes. In this paper we have tested our work for 8, 111, 21, 811 and 48 nodes. If we compare the results of CAR and E-CAR then we will found that E-CAR works fine even when CAR starts dropping data packets. After the simulation and analyzing the trace files, it has been obtained the graphs as presented.


simulationl_16.tr (83.9 GB
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Finc
ECARsimulationl_16.tr
simulationl_16.tr

```
6655 r -t 15.913804322 -Hs 5 -Hd 5 -Ni 5 -Nx 130.00
     Nw --- -Ma 13a -Md 5 -Ms 0 -Mt 800 -Is 0.255 -
     32
6656
6657  Node:5 - Congestion Density:: 6.039216
6658
6659
6660 Over Congestion
6661 Dropping Low Priority Packet at time 15.913804
6662
6663 d -t 15.913804322 -Hs 5 -Hd 13 -Ni 5 -Nx 130.0
```

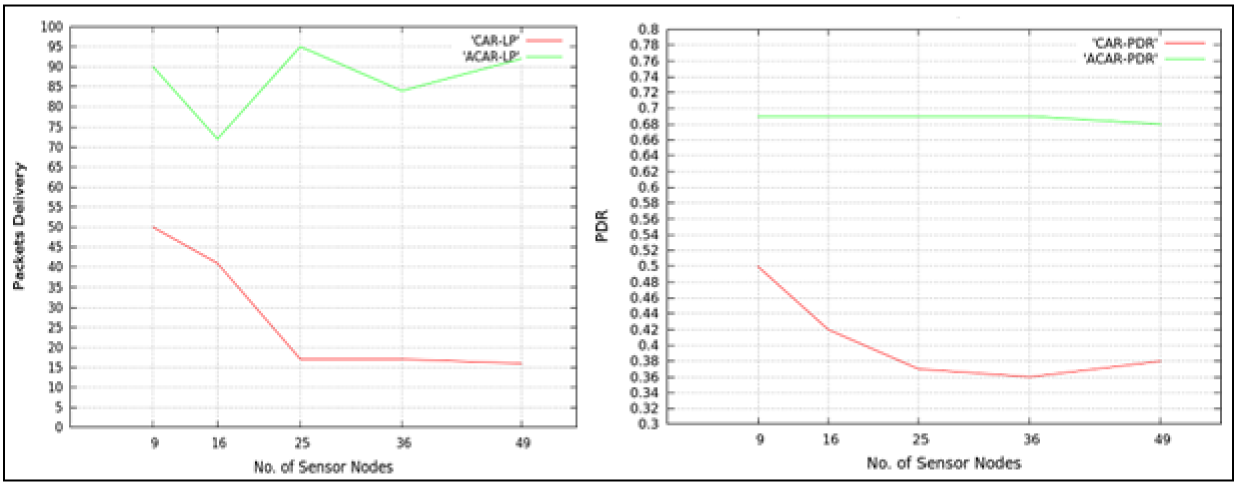
Figure 1: Dropping of packets in CAR for congestion control.


ACARsimulationl_16.tr (8
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste
ECARsimulationl_16.tr
simulationl_16.tr

```
     -Nw --- -Ma 13a -Md 7 -Ms 5 -Mt 800 -Is 7.2
     Iv 30
6049 r -t 15.895530463 -Hs 15 -Hd 15 -Ni 15 -Nx
     RTR -Nw --- -Ma 13a -Md f -Ms 7 -Mt 800 -Is
     195 -Iv 30
6050 Node 15:: I Received the Data Packet
6051
6052 -----------------------
6053  Total High Priority Packets delivered = 72
6054  Total Low Priority Packets delivered = 92
6055 -----------------------
```

Figure 2: Number of high and low priority data packets delivered in E-CAR

For the simulation, we create a square flat platform of finite dimensions for simulation. The graphs obtained show that the Packet delivery for the LP packets is much higher for E-CAR compared to CAR. The graphs also show that there is a little increase in Packet Delivery for HP packets for E-CAR compared to CAR. This may be because of routing the LP through other route than best route. Hence E-CAR achieves the best Packet Delivery compared to CAR both for HP and LP packets. E-CAR not only achieves LP packet delivery but also helps to reduce congestion in the best route from source to destination.



Hence without the loss of QoS and the effect of the congestion, the packet delivery of both the HP packets and the LP packets is to be achieved. The prediction of the congestion avoids the effect of congestion in the network and dropping of the LP packets. So the packet delivery is achieved without any conciliation in QoS.

For the simulation, we create a square flat platform of finite dimensions for simulation. The graphs obtained show that the Packet delivery for the LP packets is much higher for E-

CAR compared to CAR. The graphs also show that there is a little increase in Packet Delivery for HP packets for E-CAR compared to CAR. This may be because of routing the LP through other route than best route. Hence E-CAR achieves the best Packet Delivery compared to CAR both for HP and LP packets. E-CAR not only achieves LP packet delivery but also helps to reduce congestion in the best route from source to destination.

5. Conclusion

In this paper, we addressed data delivery issues in the presence of congestion in mobile adhoc networks. We proposed RSCARP, which is a secure differentiated routing protocol and uses data prioritization. Our extensive simulations show that as compared to AODV and AODV+PQ, RSCARP increases the fraction of HP data delivery and decrease delay and jitter for such delivery while using energy more uniformly in the deployment. RSCARP also routes an appreciable amount of LP data in the presence of congestion. Our secure sensor network communication protocol, RSCARP, offers a high level of security while requiring much less energy than previous approaches.

References

[1] C.-K. Toh, “Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks,” IEEE Communications Magazine, vol. 89, no. 11, pp. 188–147, June 2001.

[2] D. Ebenezer, Baghyalakshmi, J. Satyamurty, S.A.V. , “Low Latency and energy Efficent Routing Protocols for Mobile adhoc networks”, IEEE 207 D.O.I. 7.179/ICWCSC.207.1411892 (1-11)

[3] G. Srinivas, K. Hanumantha Rao, A. Damodhar, “Mobile adhoc networks: A Study on Congestion Routing Algorithms” IJCST, Vol. 2 Issue 2, 2011.

[4] Jason Lester Hill, “System Architecture for Mobile adhoc networks”, University of California, Berkeley, Spring 2008.

[5] J.Miguel- Alonso, C.Izu, J.A Gregario,”Improving the performance of large interconnection networks using congestion-control mechanisms", from Science Direct Performance Evaluation 111(2008) 208-211.

[6] Matulya Bansal and Gautam Barua, “Performance Comparision of Two On- Demand Routing Protocols for Mobile Ad hoc Networks” IEEE Trans. April 2004.

[7] N. Sengottiyan, Rm. Somasundaram, “A modified Routing Algorithm for Reducing Congestion in Mobile adhoc networks” European Journal of Scientific Research, Vol. 81 No. 4, 129 (2009).

[8] R. Kumar, H. Rowaihy, G. Cao, “Congestion Aware Routing in Sensor Networks” NAS-TR-00811-20011.

[9] R. Kumar, H. Rowaihy, G. Cao, “Congestion Aware Routing in Sensor Networks” NASensor Networks” IEEE transactions on mobile computing, Vol. 7 No. 7, 2008.

[10] S. Kim, O. Lee, S. Choi, “MAC-Aware Routing Metric for 802.11 Wireless Mesh Networks” IEEE, (2008).