ISSN 2047-3338

Wireless Sensor Networks: A Study on Congestion Routing Algorithms

K. Hanumantha Rao, G. Srinivas, Ankam Damodhar and M. Vikas Krishna
Sri Indu College of Engineering and Technology, Hyderabad, India

***Abstract*— Data's generated in Wireless Sensor Networks may not all alike, some data's are more important than other data's and they may have different delivery requirements. If congestion occurs in the Wireless Network, some or more important data's may be dropped. But in our project we handle this problem by addressing differentiated delivery requirements. We propose a class of algorithms that enforce differentiated routing based on the congested areas of a network and data priority. The basic protocol, called Congestion-Aware Routing (CAR), discovers the congested zone of the network that exists between high-priority data sources and the data sink and, using simple forwarding rules, dedicates this portion of the network to forwarding primarily high-priority traffic. Since CAR requires some overhead for establishing the high-priority routing zone, it is unsuitable for highly mobile data sources. To accommodate these, we define MAC-Enhanced CAR (MCAR), which includes MAC-layer enhancements and a protocol for forming high-priority paths on the fly for each burst of data. MCAR effectively handles the mobility of high-priority data sources, at the expense of degrading the performance of low-priority traffic.**

***Index Terms*– Routing, Congestion and Wireless Sensor Networks**

I. INTRODUCTION

DUE to recent technological advances, the manufacturing of small and low cost sensors became technically and economically feasible. The sensing electronics measure ambient conditions related to the environment surrounding the sensor and transform them into an electric signal. Processing such a signal reveals some properties about objects located and/or events happening in the vicinity of the sensor [1].

A large number of these disposable sensors can be networked in many applications that require unattended operations. A Wireless Sensor Network (WSN) contains hundreds or thousands of these sensor nodes. These sensors have the ability to communicate either among each other or directly to an external base-station (BS) [2]. A greater number of sensors allows for sensing over larger geographical regions with greater accuracy. Figure 1 shows the schematic diagram of sensor node components. Basically, each sensor node comprises sensing, processing, transmission, mobilizer, position ending system, and power units (some of these components are optional like the mobilizer). The same figure shows the communication architecture of a WSN [3]. Sensor nodes are usually scattered in a sensor field, which is an area where the sensor nodes are deployed. Sensor nodes

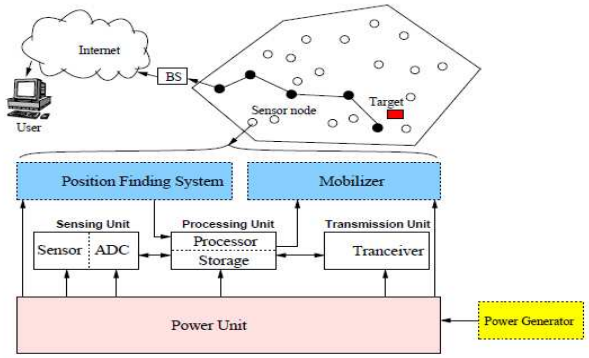

Fig. 1. Schematic diagram of sensor node components

coordinate among themselves to produce high-quality information about the physical environment. Each sensor node bases its decisions on its mission, the information it currently has, and its knowledge of its computing, communication, and energy resources. Each of these scattered sensor nodes has the capability to collect and route data either to other sensors or back to an external base station [4]. A base-station may be a fixed node or a mobile node capable of connecting the sensor network to an existing communications infrastructure or to the Internet where a user can have access to the reported data.

Networking unattended sensor nodes may have profound effect on the efficiency of many military and civil applications such as target field imaging, intrusion detection, weather monitoring, security and tactical surveillance, distributed computing, detecting ambient conditions such as temperature, movement, sound, light, or the presence of certain objects, inventory control, and disaster management. Deployment of a sensor network in these applications can be in random fashion (e.g., dropped from an airplane) or can be planted manually (e.g., alarm sensors in a facility). For example, in a disaster management application, a large number of sensors can be dropped from a helicopter. Networking these sensors can assist rescue operations by locating survivors, identifying risky areas, and making the rescue team more aware of the overall situation in the disaster area.

The rest of the paper is organized as follows: In Section 2, we elaborate on related work. In Section 3, we describes about the congestion and Routing. In Sections 4, we present the overview of the CAR & MCAR. We present conclusions and future enhancement in Section 5.

II. RELATED WORK

A. MCAR

In MCAR, each node in the network can be in one of three states, dictating whether it is a part of the con-zone or not or within the communication range of the con-zone. This last mode creates a shadow area that separates HP traffic from LP traffic.

Table 1: Shows the summary of different routing

Scheme	Summary
CAR	For static or nearly-static conzone and long-lived high priority flows
CAR+	Conzone nodes drop all low priority data
CAR++	Conzone nodes and neighbors of critical area drop all low priority data
MCAR	For mobile high priority data sources or short-lived high priority flows

B. Dynamic Con-Zone Discovery

Nodes discover if they are on the con-zone by using the con-zone discovery mechanism. After building the HiNet, the next task is to dynamically discover the con-zone.

The con-zone is formed when one area is generating HP data. Refer to this area as the critical area. This con-zone 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. The con-zone can be discovered and destroyed either from the critical area nodes to the sink or vice versa. The con-zone discovery algorithms allow nodes, in a distributed fashion, to determine if they are on a potentially congested path between the critical area and the sink. If they are, they mark themselves as "on con-zone." The con-zone discovery schemes are summarized in Figure. For brevity, only present con-zone discovery from the critical area to the sink in detail.

In this case, critical area nodes detect an event that triggers discovery. A con-zone must be then discovered from that neighborhood to the sink for the delivery of HP data. To do this, critical area nodes broadcast "discover con-zone to sink" (To Sink) messages. This message includes the ID of the source and its depth and is overheard by all neighbors. The depth is included here to ensure that nodes do not respond to the To Sink messages heard from their parents.

When a node hears more than distinct To Sink messages coming from its children, it marks itself as on con-zone and propagates a single To Sink message. Since the depth and neighborhood size can vary for different nodes, is set accordingly. Setting correctly for different depths ensures that the con-zone is of an appropriate width. As becomes smaller,

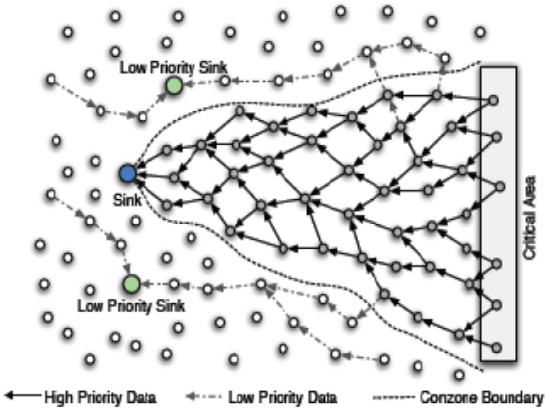

Fig. 8. Shows the dynamic con-zone discoveries

the con-zone becomes wider. Depth must also be taken into account, because if it is the same for different depths, the con-zone will become very narrow as it approaches the sink.

Note that due to the assumption of uniform deployments, the neighborhood size is related to the number of children by a constant factor. Number of children, but use the neighborhood size instead. An important goal of the con-zone discovery algorithm into split the parents and siblings (nodes with the same depth) in the HiNet into on-con-zone and off-con-zone neighbors. Initially, all parents and siblings are marked as off con-zone. Since a node will forward a To Sink message only if it becomes on con-zone, when a node hears such a broadcast from its parent(s) or sibling(s), it marks that neighbor as on con-zone.

C. Differentiated Routing

Once the con-zone is discovered, our next task is to route high priority data on the con-zone and route the low priority data off the con-zone. Since the critical area is a part of the con-zone, all high priority data will be generated inside the con-zone. Routing of high priority data in this case is very simple; a node always forwards the data to one of its parents.

This parent is chosen randomly from the parent list to balance the load between them. This continues until the sink is reached. If for some reason the links to all parents are broken, because of node failures for example, a node will forward the data to a sibling which is on the con-zone. If that is impossible it will forward the data to any of its neighbors hoping that it can return to an on-con-zone node. All low priority data generated inside the conzone must be routed out. There are two cases to consider.

An on-con-zone node that generates or receives low priority data has a parent or sibling that is off-con-zone.

When an on-con-zone node gets a low priority message it forwards it to an off-con-zone parent, if there are any. Otherwise the low priority data is forwarded to an off-con-zone sibling (which is a node with the same depth). If there are no parents or siblings that are off-con-zone,

After discovering the con-zone, the sink sends a message through the con-zone which contains the coordinates of a line that cuts the con-zone 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 con-zone it lies and hence route low priority data to the parent that is closest to the con-zone boundary, farthest from the line. With the assumption of uniform deployment density, this ensures that all low priority data generated inside the con-zone is routed out efficiently and along the shortest path.

D. Transmission Range

The primary challenge in implementing MCAR involved the strictly modular design of Tiny OS. Because MCAR relies on priority information from the application layer and alters both the routing and MAC layers, it was necessary to find clean ways to pass information between the layers. But MCAR's mechanisms work in a top-down manner (i.e., the adaptations are driven by the application priority settings), only these priorities need to be exposed to all layers. For example, any route setup packets for an HP flow must be assigned an HP, or they risk being dropped. However, route setup in many standard protocols is not tagged with flow information. Therefore, application priorities must be used at the routing layer, and all routing mechanisms used to service an HP flow must themselves be HP. While such changes in protocols are small, in terms of code size, they are critical for protocol correctness.

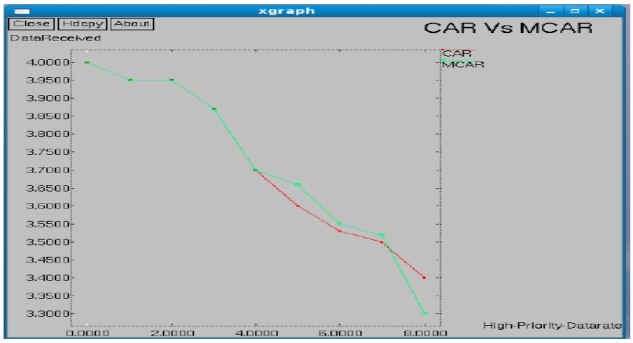

Fig. 9. Graph for performance of high-priority data

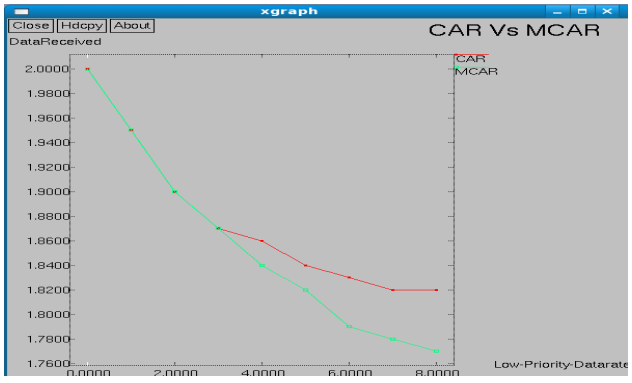

Fig. 10. Grapgh for performance of low-priroty data

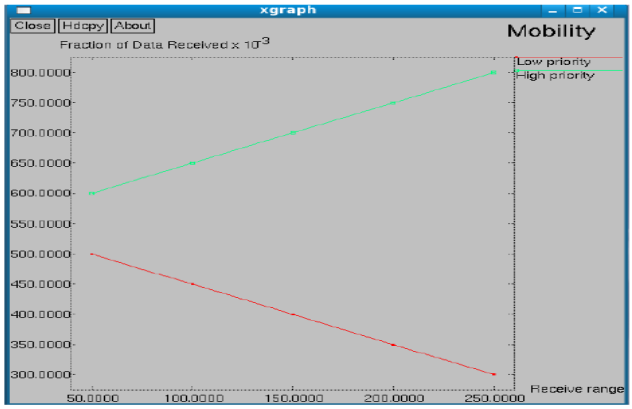

Fig. 11. Grapgh for performance of low-priroty/high-priorty data

III. CONGESTION

A. Types of Congestion Control Schemes

(i) Open Loop Congestion Control Schemes

- Traffic Filtering Schemes (use accept/reject rules)
- Traffic Scheduling Schemes

(ii) Closed Loop Congestion Control Schemes

- Uni-Variable Feedback based schemes
- Multi-Variable Feedback based schemes

B. Congestion Metrics

- Average/Mean Queue Length
- Average number or percentage of lost/discarded packets
- Number of retransmitted packets those had to be sent again because of Transmitter's Timeout
- Average/Mean Delay in Packet Delivery

C. Congestion Control Strategies

- Congestion control by regulating admission of Packets/Cells
- Congestion control by regulating traffic based on traffic-Type/traffic-rate (packet rate/cell rate/bit rate etc) analysis
- Congestion control by admission-time resource reservation
- Congestion control by threshold monitoring and message passing
- Congestion control by preferential restraint (in research stage)
- Congestion control by Ostrich algorithm (debatable)
- Congestion control by supervised blocking/rerouting (under investigation)

D. Classification of congestion control algorithms

There are many ways to classify congestion control algorithms:

By the type and amount of feedback received from the network: Loss; delay; single-bit or multi-bit explicit signals

By incremental deploy ability on the current Internet: Only sender needs modification; sender and receiver need modification; only router needs modification; sender, receiver and routers need modification.

By the aspect of performance it aims to improve: high bandwidth-delay product networks; lossy-links; fairness; advantage to short flows; variable-rate links

By the fairness criterion it uses: max-min, proportional, "minimum potential delay".

E. Routing

Routing is the act of moving information across an internetwork from a source to a destination. Along the way, at least one intermediate node typically is encountered. Routing is often contrasted with bridging, which might seem to accomplish precisely the same thing to the casual observer. The primary difference between the two is that bridging occurs at Layer 2 (the link layer) of the OSI reference model.

F. Routing Components

Routing involves two basic activities: determining optimal routing paths and transporting information groups (typically called packets) through an internetwork. In the context of the routing process, the latter of these is referred to as packet switching. Although packet switching is relatively straightforward, path determination can be very complex.

G. Routing Algorithms

Routing algorithms can be differentiated based on several key characteristics. First, the particular goals of the algorithm designer affect the operation of the resulting routing protocol. Second, various types of routing algorithms exist, and each algorithm has a different impact on network and router resources. Finally, routing algorithms use a variety of metrics that affect calculation of optimal routes. The following sections analyze these routing algorithm attributes.

1. Path Determination

Routing protocols use metrics to evaluate what path will be the best for a packet to travel [7]. A metric is a standard of measurement, such as path bandwidth, that is used by routing algorithms to determine the optimal path to a destination. Routing algorithms fill routing tables with a variety of information. Destination/next hop associations tell a router that a particular destination can be reached optimally by sending the packet to a particular router representing the “next hop” on the way to the final destination. Routing tables also can contain other information, such as data about the desirability of a path. Routers compare metrics to determine optimal routes, and these metrics differ depending on the design of the routing algorithm used. Routers communicate with one another and maintain their routing tables through the transmission of a variety of messages. The routing update message is one such message that generally consists of all or a portion of a routing table. By analyzing routing updates from all other routers, a router can build a detailed picture of network topology.

2. Design Goals

Routing algorithms often have one or more of the following design goals:

Optimality: refers to the capability of the routing algorithm to select the best route, which depends

on the metrics and metric weightings used to make the calculation.

Simplicity and low overhead: The routing algorithm must offer its functionality efficiently, with a minimum of software and utilization overhead.

Robustness and Stability: Routing algorithms must be *robust*, which means that they should perform correctly in the face of unusual or unforeseen circumstances, such as hardware failures, high load conditions, and incorrect implementations.

Rapid Convergence: Convergence is the process of agreement, by all routers, on optimal routes. When a network event causes routes to either go down or become available, routers distribute routing update messages that permeate networks.

Flexibility: Routing algorithms should be flexible, which means that they should quickly and accurately adapt to a variety of network circumstances.

H. Routing Metrics

Routing tables contain information used by switching software to select the best route. Sophisticated routing algorithms can base route selection on multiple metrics, combining them in a single (hybrid) metric. The following metrics have been used:

Path length, Reliability, Delay, Bandwidth, Load, Communication cost.

IV. CONGESTION AWARE ROUTING

CAR comprises three steps; i) HP network formation, ii) Conzone discovery and iii) Differentiated routing.

The combination of these functions segments the network into on-conzone and off-conzone nodes. Only HP traffic is routed by on-conzone nodes. The protocol specifically accommodates LP traffic, albeit with less efficient routes than HP traffic. For the purposes of this discussion, assume that there is one HP sink and a contiguous part of the network (critical area) that generates HP data in the presence of network wide background LP traffic.

Nodes are location aware and densely deployed with uniform distribution. Since nodes in the scenario send all HP data to a single sink, tree-based routing, with the HP sink being the root, is most appropriate. The tree-based routing schemes suffer from congestion, especially if the number of messages generated at the leaves is high.

This problem becomes even worse when a mixture of LP and HP traffic travel through the network. Therefore, even when the rate of HP data is relatively low, the background noise created by LP traffic will create a conzone that spans the network from the critical area to the HP sink. Due to this congestion, service provided to HP data may degrade, and nodes within this area may die sooner than others, leading to only suboptimal paths being available for HP data, or a

network partition may result, isolating the sink from the critical area.

A. High-Priority Routing Network Formation

After the deployment of sensor nodes, the HP data collection center (the 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. Also, based on the locations of events that can occur during the lifetime of the network, different nodes may constitute the critical area. Since all HP data is destined to a single sink, the HiNet is based on a minimum distance spanning tree rooted at the sink .this structure ensures that all nodes have shortest path routes to the sink.

However, instead of every node having a single parent, as in other tree-based schemes, and allow nodes to have multiple parents. A node that has multiple neighbors with depths (the number of hops to the sink) less than its own considers them all as parents. Leverage this property to support multipath forwarding, thus providing load balancing and making the routing network more resilient to failures.

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.

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

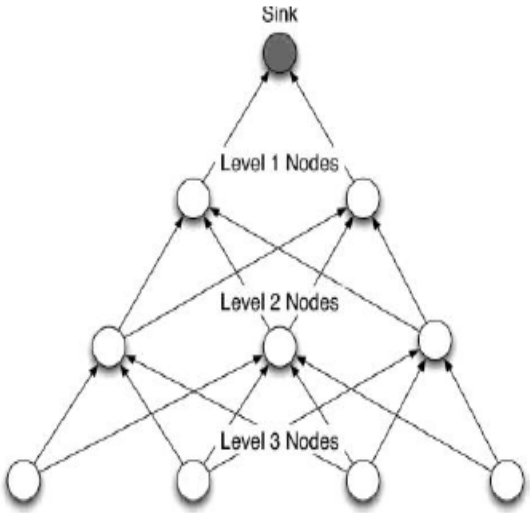

Fig. 2. In a dense deployment, multiple nodes can be parents of a node. Each parent lies on a different shortest path route to the sink. This structure is used for shortest multipath routing

Conzone Discovery From Critical area to sink:
if *node x receives ToSink from child* c_i **then**
if *On_conzone is==FALSE* ***then***
if ToSink_received > α_x **then**
On.Conzone =TRUE
if x is not sink **then**
broadcast ToSink with d_z
else
ToSink_received ++
else if node x receives ToSink from parent p_j **then**
P_{off} - ={p_j}; P_{on}+={p_j}
else if node x receivers ToSink from sibling s_l **then**
S_{off}-={s_l};S_{on}+={s_l}

Conzone Discovery From Sink To Critical Area:
if node x receives FromSink from parent then p_i **then**
P_{off} - ={p_j}; P_{on}+={p_j}
if On.Conzone = = FALSE **then**
if x has a crictical area child c_j ϵ then critical area **then**
On.Conzone =TRUE
if x is not critical area node **then**
Broadcast FromSink with $depth_x$
else if node x receives FromSink from sibling sl **then**
S_{off}-={s_l}; S_{on}+={s_l}

Fig. 3. Example of the algorithms is taken from ref [15]

Conzone discovery algorithms in CAR for node x.
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: $Children=\{c_1, c_2, ...c_n\}$
Node's on-conzone status: *On-Conzone=FALSE*
ToSink messages received: *ToSink_received=0*
ToSink threshold: $\alpha_x = \beta_{dz}.d_x.n_x$

Conzone Discovery From Critical area to sink:
if *node x receives ToSink from child* c_i **then**
if *On_conzone is==FALSE* ***then***
if ToSink_received > α_x **then**
On.Conzone =TRUE
if x is not sink **then**
broadcast ToSink with d_z
else
ToSink_received ++
else if node x receives ToSink from parent p_j **then**
P_{off} - ={p_j}; P_{on}+={p_j}
else if node x receivers ToSink from sibling s_l **then**
S_{off}-={s_l};S_{on}+={s_l}

Fig. 4. Example of the algorithms is taken from ref [15]

Routing algorithm for CAR for LP and HP data inside the conzone.

Routing Low Priority Data:
If $P_{off} \neq \{\}$ **then**
  send data to any $p \in P_{off}$
else if $\exists$ a sibling $s \in S_{0ff}$ **then**
  send data to s
else
  Send data to the farthest parent p from diving line
Routing High Priority Data:
If $P_{on} \neq \{\}$ **then**
send data to any $p \in P_{on}$
else if $\exists$ a sibling $s \in S_{on}$ **then**
send data to s
else
send data to any $u \in P_{off} U S_{off}$

Fig. 5. Example of the algorithms is taken from ref [15]

one less than its own, then the source of the message is added as a parent. In this case, the message is not rebroadcast. If a node receives a Build HiNet message with a depth value less than that of its parent's depth, it updates its own value to the received value.

B. Mac-Enhanced Congestion Aware Routing (MCAR)

A combined MAC and routing scheme designed to support situations in which critical events may move or the sensors generating HP data may move. Though conzone discovery is dynamic in CAR, the overhead required to maintain the HiNet in a dynamic environment may be prohibitive. As a result, use a lightweight dynamic differentiated routing mechanism to accommodate mobile data sources.

MCAR is based on MAC-layer enhancements that enable the formation of a conzone on the fly with each burst of data. The trade-off is that it effectively preempts the flow of LP data, thereby seriously degrading its service. Unlike CAR, MCAR does not form an HP network. Instead, HP paths are dynamically created, since the sources (or the sinks) are expected to be mobile.

Thus, MCAR discovers the conzone while discovering the paths from HP sources to the sink. The enhanced MAC-layer of MCAR uses an RTS/CTS protocol that is augmented to carry information about the priority level of the data being transferred. Each RTS and CTS packet is tagged with a priority level. During channel contention, if a node has HP data to send and overhears an LP RTS, it jams the channel with an HP CTS, causing nodes forwarding LP data to back off. Furthermore, if a node with LP data overhears an HP RTS or CTS, it will back off the channel, as described in the following section. The prioritized RTS/CTS messages in highly congested networks may be dropped. The extent of overhead experienced depends on the relative size of the RTS/CTS packets and the data packets. In sensor networks, data packet sizes are not large enough to justify the cost of RTS/CTS exchange to guard every packet.

Hence, 802.11e is unsuitable for sensor networks. MCAR uses a silencing mechanism that does not require preempting all LP data transmissions in the neighborhood for each HP data to be sent. Rather, MCAR silences the conzone and its neighborhood during route discovery and/or maintenance.

Though the cost of an RTS/CTS exchange for each data packet may be considerable for a sensor network, even S-MAC a widely used MAC scheme for sensor networks, uses one RTS/CTS exchange for a collection of message fragments. Similarly, the cost of RTS/CTS imposed by MCAR is not prohibitive, since it uses these RTS/CTS packets only during the route discovery/maintenance phase. Hence, the scalability of the RTS/CTS overhead for MCAR is not an issue.

C. MCAR State Machines

LP mode: In this mode, nodes forward LP data. All nodes in the network are initially in the LP mode. Upon receiving or overhearing an LP packet, nodes remain in the LP mode and, if appropriate, forward any data. If a node in the LP mode overhears an HP packet, it transitions to the shadow mode. Finally, upon receiving an HP event that needs to be forwarded (either because it sensed an HP event or because it was chosen as the next hop toward the sink), a node transitions to the HP mode.

D. HP Mode: Nodes in the path of HP data are in the HP mode. Upon transitioning to this state, the node sets two timers: a received timer and an overhearing timer. The values for these timers should be on the order of twice the expected inter arrival delay of HP data. If a node in this mode receives an HP transmission, it begins channel contention by using our modified RTS/CTS protocol and forwards the data. It resets its received and overhearing timers and remains in the HP mode. Upon overhearing HP data, the node resets its overhearing timer only and stays in the HP mode.

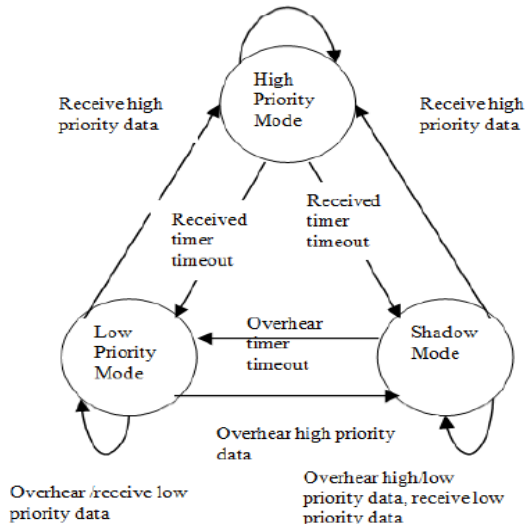

Fig. 6. Different state transition diagram

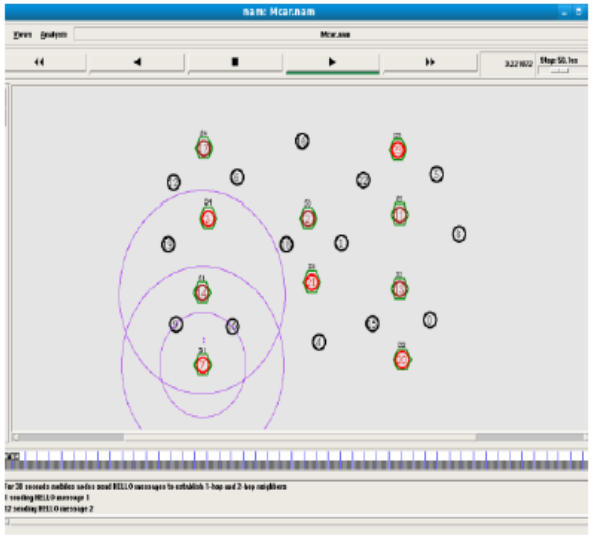

Fig. 7. Different shadow modes

If a node in the HP mode overhears or receives an LP RTS, it sends a jamming HP CTS to clear the channel of LP data and to announce the existence of an HP path and stays in the HP mode.

If the received timer expires, the node transitions to the shadow mode, maintaining the value of its overhearing timer. While this is the normal exit out of the HP mode, if both the received timer and overhearing timer expire at the same time, the node transitions back to the LP mode.

Shadow mode: Nodes in this state (Fig. 8) are within the communication range of HP traffic but not on a forwarding path. Nodes in this state suppress LP traffic, thus preventing it from interfering with HP traffic in the network. Upon overhearing an HP packet, the node resets its overhearing timer and stays in this state. A node transitions to the HP mode upon receiving an HP packet itself.

If a node in the shadow mode overhears an LP packet, it stays in the shadow mode and takes no action. If the node is the intended recipient of the LP data, it silently discards the packet and stays in the shadow mode. It should be pointed out that this is an aggressive action to maximize the service given to HP data. Finally, if the overhearing timer expires the node transitions to the LP mode.

Routing: Route discovery is performed dynamically at the time of HP event detection. Essentially, MCAR performs on-demand route discovery similar to schemes like AODV. The route discovery and reply packets are marked according to the priority of impending data, causing nodes along the route for HP data to transition to the HP mode. Once the route is built, HP data flows along this path. In the event of a route break due to node failure or mobility, route recovery is performed, again using HP control packets. Nodes on segments of the old route will transition back to the LP mode as their timers expire, and LP flows that were not forwarded can now be transmitted.

Only nodes in the LP mode forward LP data, including any LP route requests. The routing of this data can be performed using any routing mechanism and is orthogonal to the routing mechanisms used by MCAR. Nodes in the HP or the shadow mode drop LP data. Hence, there is no need to route LP data out of the HP zone in MCAR. As a result, MCAR is more aggressive in dropping LP data and eliminates all competition for the shared channel among the LP and HP packets. This is one of the trade-offs between CAR and MCAR.

Although both schemes support HP data delivery, CAR is able to route LP traffic out of the con-zone, while MCAR cannot. CAR requires the formation of the HiNet, which incurs higher overhead than the dynamic path establishment of MCAR. CAR is more permissive of LP traffic than MCAR: it allows nodes that would be in the shadow mode in MCAR to forward LP data. MCAR, on the other hand, performs more similarly to CAR++ in this respect, limiting the use of nodes in the con-zone to only HP data. Section 4 quantifies these trade-offs through simulation studies.

In MCAR, nodes discover if they are on the con-zone by using the con-zone discovery explained in the following. Like CAR, this con-zone discovery is triggered when an area starts generating HP data. For the con-zone to be discovered dynamically, MCAR uses two timers to regulate when a node decides it is no longer part of the HP path. One timer, called the overhearing timer, monitors how long it has been since the last HP packet was heard. This timer is used to control nodes in the communication range of the con-zone but that are not necessarily involved in forwarding the packets.

The overhearing timer is reset any time an HP packet is overheard or any time an HP packet is received (since nodes involved in forwarding packets are clearly within the communication range of nodes transmitting those packets). The second timer, called the received timer, controls nodes either generating or forwarding HP data.

V. CONCLUSION AND FUTURE WORK

In this paper, we addressed data delivery issues in the presence of congestion in wireless sensor networks. We proposed CAR, which is a differentiated routing protocol and uses data prioritization. We also develop MCAR, which deals with mobility and dynamics in the sources of HP data. Both CAR and MCAR support effective HP data delivery in the presence of congestion. CAR is better suited for static networks with long-duration HP floods. For HP traffic and/or mobile HP sources, MCAR is a better fit. To better serve HP data, on-con-zone nodes stop generating or forwarding any LP data. We call this enhancement CAR+. We disable generating and forwarding of LP data in all nodes that are within the communication range of any critical area node. Since nodes know their neighbors and their status, once a node discovers that one of its neighbors is on the critical area, it disables generating and forwarding of any LP data. We call this enhancement CAR++.

REFERENCES

[1] Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," IEEE Communications Magazine, Volume: 40 Issue: 8, pp.102-114, August 2002.

[2] Perrig, R. Szewzyk, J.D. Tygar, V. Wen, and D. E. Culler, "SPINS: security protocols for sensor networks". Wireless Networks Volume: 8, pp. 521-534, 2000.

[3] S. Hedetniemi and A. Liestman, \A survey of gossiping and broadcasting in communication networks", IEEE Networks, Vol. 18, No. 4, pp. 319-349, 1988.
[4] Anquetil, N. and Lethbridge, T., "Assessing the Relevance of Identifier Names in a Legacy Software System", in Proceedings of Annual IBM Centers for Advanced Studies Conference (CASCON'98), December 1998, pp. 213-222.
[5] Stephan Olariu, “Information assurance in wireless sensor networks”, Sensor network research group, Old Dominion University.
[6] C. Chong and S. P. Kumar, “Sensor Networks: Evolution, Opportunities, and Challenges”, in Proceedings of the IEEE, vol. 91, no. 8, Aug. 2003.
[7] Karp and H. T. Kung, “GPSR: greedy perimeter stateless routing for wireless networks”, in Mobile Computing and Networking, 2000, pp. 243–254.
[8] Holger Karl and Andreas Willig. “Protocols and architectures for Wireless Sensor Networks”, Wiley, 2005, ISBN: 0470095105.
[9] K. Sohrabi et al., “Protocols for self-organization of a wireless sensor network”, IEEE Personal Communications 7 (5) (2000) 16–27.
[10] F. Ye, H. Luo, J. Cheng, S. Lu, L. Zhang, \A Two-tier data dissemination model for large-scale wireless sensor networks", proceedings of ACM/IEEE MOBICOM, 2002.
[11] N. Bulusu, J. Heidemann, D. Estrin,\GPS-less low cost outdoor localization for very small devices", Technical report 00-729, Computer science department, University of Southern California, April, 2000.
[12] A. Savvides, C-C Han, and M. Srivastava, “Dynamic grained localization in Ad-Hoc networks of sensors," Proceedings of the Seventh ACM Annual International Conference on Mobile Computing and Networking (MobiCom), July 2001, pp. 166-179.
[13] V. Rodoplu and T. H. Meng, “Minimum Energy Mobile Wireless Networks", IEEE Journal Selected Areas in Communications, vol. 17, no. 8, Aug. 1999, pp. 133344.
[14] Wei Yen, Ching-Wei Chen and Cheng-hsiang Yang, “Single Gossiping with Directional Flooding Routing Protocol in Wireless Sensor Networks”, in Proceedings IEEE, 2008.
[15] Mitigating Performance Degradation in Congested Sensor Networks,Raju Kumar, Student Member, IEEE, Riccardo Crepaldi, Student Member, IEEE,Hosam Rowaihy, Student Member, IEEE, Albert F. Harris III, Member, IEEE,Guohong Cao, Senior Member, IEEE, Michele Zorzi, Fellow, IEEE, and Thomas F. La Porta, Fellow, IEEE

K. Hanumantha Rao, working as Asst. Prof. at Sri Indu College of Engineering & Technology, Hyderabad, He has guided many PG level and Engineering students, areas of interest are Operating System, Mobile Computing and Information Security.

G. Srinivas, pursuing M.Tech CS at Sri Indu College of Engg & Tech from JNTU Hyderabad, areas of interest are Network Security, Wireless Sensor Networks and Mining.

Ankam Damodhar pursuing M.Tech CS at Sri Indu College of Engg. & Tech. from JNTU Hyderabad, areas of interest are Information Security, Mobile Computing, and Data Warehousing.

M. Vikas Krishna pursuing M.Tech CS at Sri Indu College of Engg. & Tech. from JNTU Hyderabad, areas of interest are Information Security, Mobile Computing, Data Warehousing and Mining.