Congestion Control in Wireless Sensor Networks

N.Premalatha1[†] and Dr.A.M.Natarajan2[††],

[1]Senior Lecturer, Dept. Of CSE, Kongu Engineering College, Perundurai, Erode
[2]Chief Executive & Professor, Bannariamman Institute of Technology, Sathyamangalam

Abstract

Nowadays, there is a huge increase of handled devices. Laptops, mobile phones and PDAs take an important place in the everyday life. Hence, the challenge is now to make all these devices communicate together in order to build a network. Obviously, this kind of networks has to be wireless. Indeed, the wireless topology allows flexibility and mobility. In this context, the idea of ad hoc networks was developed. Network congestion occurs when offered traffic load exceeds available capacity at any point in a network. In wireless sensor networks, congestion causes overall channel quality to degrade and loss rates to rise, leads to buffer drops and increased delays (as in wired networks), and tends to be grossly unfair toward nodes whose data has to traverse a larger number of radio hops.

Key words:
Wireless sensor networks, congestion control, flow control, rate limiting, network performance

1. Introduction

Wireless Sensor Network

A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations. The development of wireless sensor networks was originally motivated by military applications such as battlefield surveillance. However, wireless sensor networks are now used in many industrial and civilian application areas, including industrial process monitoring and control, machine health monitoring, environment and habitat monitoring, healthcare applications, home automation, and traffic control.

The applications for WSNs are many and varied, but typically involve some kind of monitoring, tracking, and controlling. Specific applications for WSNs include habitat monitoring, object tracking, nuclear reactor control, fire detection, and traffic monitoring. In a typical application, a WSN is scattered in a region where it is meant to collect data through its sensor nodes. A number of WSN deployments have been done in the past in the context of environmental monitoring. A sensor node, also known as a 'mote', is a node in a wireless sensor network that is capable of performing some processing, gathering sensory information and communicating with other connected nodes in the network.

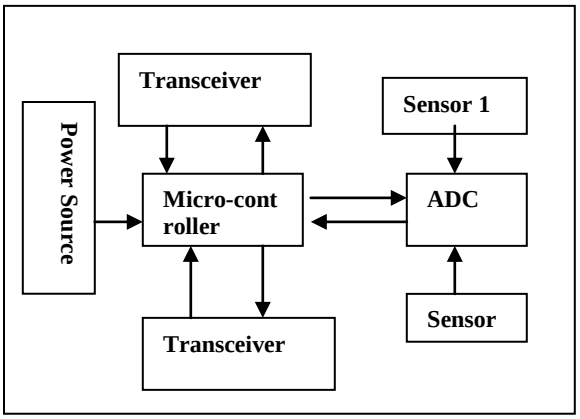

Fig. 1 Architecture of Sensor node.

The main components of a sensor node as seen from the figure are microcontroller, transceiver, external memory, power source and one or more sensors. Microcontroller performs tasks, processes data and controls the functionality of other components in the sensor node. Microcontrollers are most suitable choice for sensor node.

From an energy perspective, the most relevant kinds of memory are on-chip memory of a microcontroller and FLASH memory off-chip RAM is rarely if ever used. Flash memories are used due to its cost and storage capacity. Memory requirements are very much application dependent. Two categories of memory based on the purpose of storage a) User memory used for storing application related or personal data. b) Program memory used for programming the device. This memory also contains identification data of the device if any. Power consumption in the sensor node is for the Sensing, Communication and Data Processing. More energy is required for data communication in sensors.

2.1 Existing System

Manuscript received March 5, 2010
Manuscript revised March 20, 2010

Congestion becomes an important problem. Congestion may lead to indiscriminate dropping of data (i.e., high-priority (HP) packets may be dropped while low-priority (LP) packets are delivered). It also results in an increase in energy consumption to route packets that will be dropped downstream as links become saturated. As nodes along optimal routes are depleted of energy, only nonoptimal routes remain, further compounding the problem. To ensure that data with higher priority is received in the presence of congestion due to LP packets, differentiated service must be provided.

Congestion becomes worse when a particular area is generating data at a high rate. This may occur in deployments in which sensors in one area of interest are requested to gather and transmit data at a higher rate than others. In this case, routing dynamics can lead to congestion on specific paths. These paths are usually close to each other, which lead to an entire zone in the network facing congestion. This zone, essentially an extended hotspot, is the congestion zone (conzone).

Drawbacks of Existing system

- AODV does not support for routing discovery
- Route Maintenance cannot be performed.

2.2 Proposed System

The various modules involved in this process are as follows,

- Congestion Aware Routing (CAR)
- MAC-Enhanced Congestion Aware Routing

Shadow mode (MCAR)

2.2.1 Congestion Aware Routing

CAR comprises three steps

- HP network formation
- Conzone discovery
- 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.

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 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.

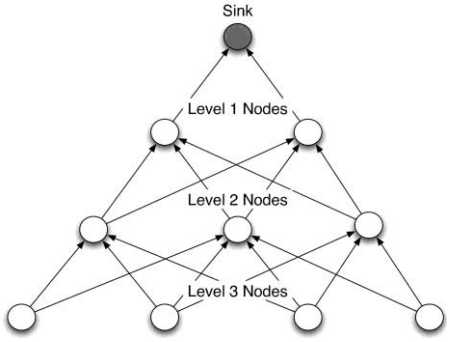

Fig. 2 : In a dense deployment, multiple nodes can be parents of a node. Each parent lies on a different shortest path route to the sink. This structure is used for shortest multipath routing

Conzone discovery 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}+=$\{s_l\}$

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}+=$\{s_l\}$

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 ϵ P_{off}
else if ∋ a sibling s ϵ 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 ϵ P_{on}
else if ∋ a sibling s ϵ S_{on} **then**
send data to s
else
send data to any u ϵ P_{off} U S_{off}

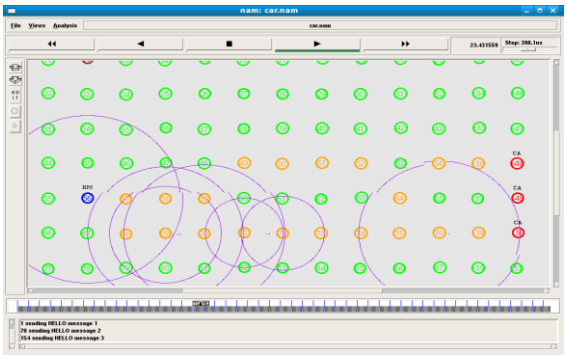

Fig. 3 Data transfer in static nodes

Fig. 4 Data transfer in mobility nodes

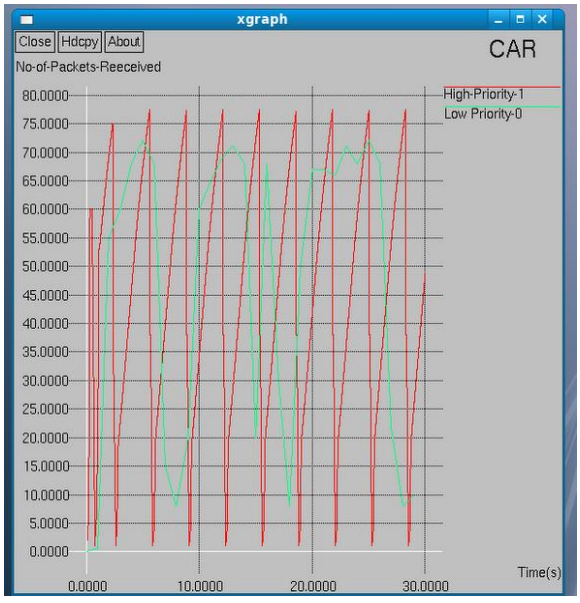

Fig.5 Xgraph of CAR

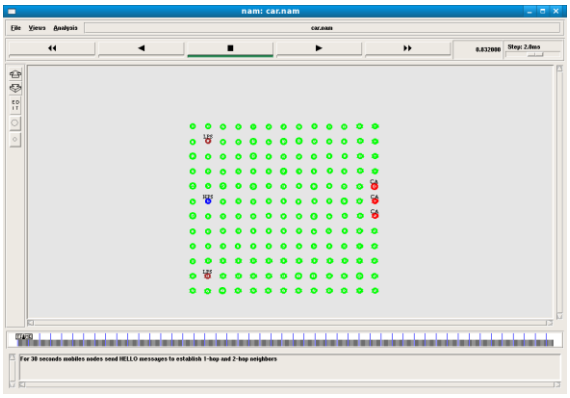

Fig. 6 CAR Animator windows

2.2.2 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.

MCAR State machines:

Fig.7 gives the different state transition diagram.

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.

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 interarrival 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.

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.

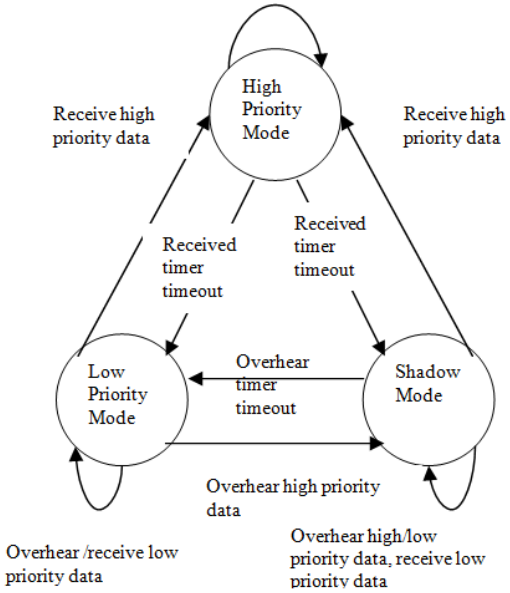

Fig. 7. MCAR state machines

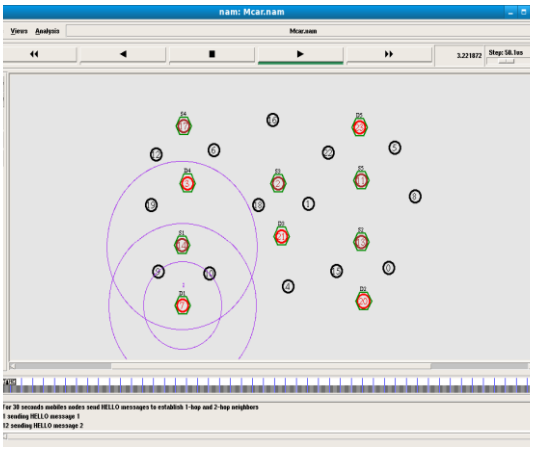

Fig 8. MCAR animated windows

Shadow mode: Nodes in this state 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 conzone, 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 conzone to only HP data. Section 4 quantifies these trade-offs through simulation studies.

In MCAR, nodes discover if they are on the conzone by using the conzone discovery explained in the following. Like CAR, this conzone discovery is triggered when an area starts generating HP data. For the conzone 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 conzone 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.

In MCAR, each node in the network can be in one of three states, dictating whether it is a part of the conzone or not or within the communication range of the conzone.

This last mode creates a shadow area that separates HP traffic from LP traffic.

Table 1: Summary of schemes

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

3. Problem Solution

3.1. Dynamic Conzone Discovery

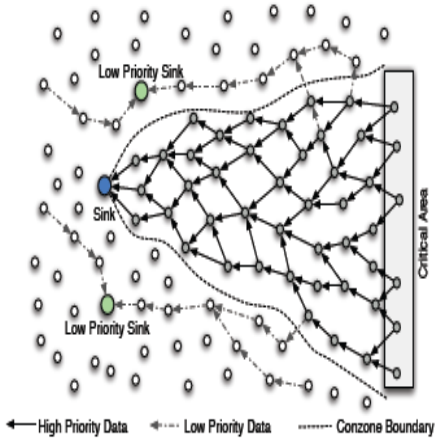

Fig. 8 Dynamic conzone discoveries

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

The conzone is formed when one area is generating HP data. Refer to this area as the critical area. 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. The conzone can be discovered and destroyed either from the critical area nodes to the sink or vice versa. The conzone 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 conzone." The conzone discovery schemes are summarized in Figure. For brevity, only present conzone discovery from the critical area to the sink in detail.

In this case, critical area nodes detect an event that triggers discovery. 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" (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 conzone 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 conzone is of an appropriate width. As becomes smaller, the conzone becomes wider. Depth must also be taken into account, because if _ is the same for different depths, the conzone 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 conzone discovery algorithm into split the parents and siblings (nodes with the same depth) in the HiNet into on-conzone and off-conzoneneighbors.

Initially, all parents and siblings are marked as off conzone. Since a node will forward a To Sink message only if it becomes on conzone, when a node hears such a broadcast from its parent(s) or sibling(s), it marks that neighbor as on conzone.

3.2 Differentiated Routing

Once the conzone is discovered, our next task is to route high priority data on the conzone and route the low priority data off the conzone. Since the critical area is a part of the conzone, all high priority data will be generated inside the conzone. 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 conzone.

If that is impossible it will forward the data to any of its neighbors hoping that it can return to an on-conzone node. All low priority data generated inside the conzone must be routed out.

There are two cases to consider.

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

- When an on-conzone node gets a low priority message it forwards it to an off-conzone parent, if there are any. Otherwise the low priority data is forwarded to an off-conzone sibling (which is a node with the same depth).If there are no parents or siblings that are off-conzone,
- 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 route low priority data to the parent that is closest to the conzone boundary, farthest from the line. With the assumption of uniform deployment density, this ensures that all low priority data generated inside the conzone is routed out efficiently and along the shortest path.

4.Project Decomposition

4.1 Path-Finding-Process: Route Request & Route Reply

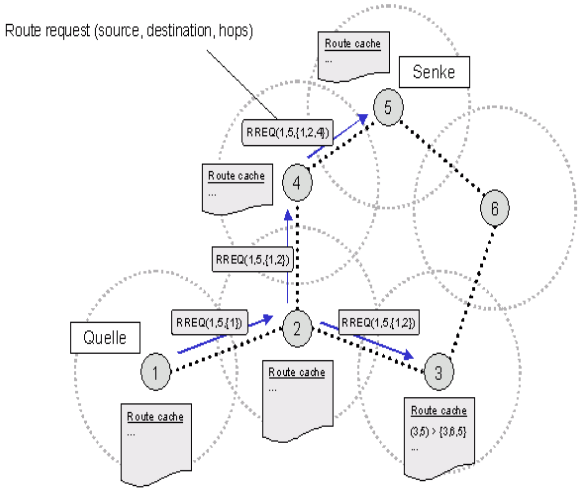

Fig. 10 Route request

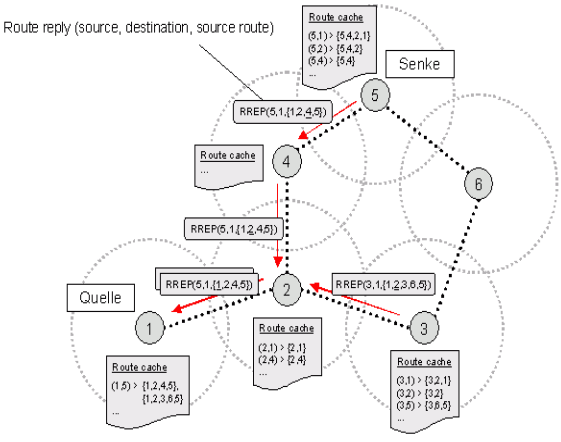

Fig.11. Route reply

4.2 Route Maintenance

In DSR every node is responsible for confirming that the next hop in the Source Route receives the packet. Also each packet is only forwarded once by a node (hop-by-hop routing). If a packet can't be received by a node, it is retransmitted up to some maximum number of times until a confirmation is received from the next hop.

Only if retransmission results then in a failure, a Route Error message is sent to the initiator that can remove that Source Route from its Route Cache. So the initiator can check his Route Cache for another route to the target. If there is no route in the cache, a Route Request packet is broadcasted.

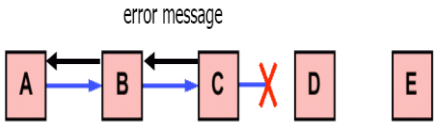

Fig.12. Route error message

1. If node C does not receive an acknowledgement from node D after some number of requests, it returns a Route Error to the initiator A.
2. As soon as node receives the Route Error message, it deletes the broken-link-route from its cache. If A has another route to E, it sends the packet immediately using this new route.
3. Otherwise the initiator A is starting the Route Discovery process again.

Advantages

Reactive routing protocols have no need to periodically flood the network for updating the routing tables like table-driven routing protocols do. Intermediate nodes are able to utilize the Route Cache information efficiently to reduce the control overhead. The initiator only tries to find a route (path) if actually no route is known (in cache). Current and bandwidth saving because there are no hello messages needed (beacon-less).

Disadvantages

The Route Maintenance protocol does not locally repair a broken link. The broken link is only communicated to the initiator. The DSR protocol is only efficient in MANETs with less then 200 nodes. Problems appear by fast moving of more hosts, so that the nodes can only move around in this case with a moderate speed. Flooding the network can cause collusions between the packets. Also there is always a small time delay at the beginning of a new connection because the initiator must first find the route to the target.

Varying Transmission Range

In this group of simulations, the transmission ranges were varied between 90, 130, 170, and 210 m. As the transmission range increases, the number of hops from the edge of the network to the sink decreases from 6 to 3. The LP data rate of each node, other than the critical area nodes and the sinks, was set to 0.5 pps, while the HP data rate of

critical area nodes was set to 30 pps. These simulations show the gains of CAR schemes and MCAR as the node density of a deployment increases.

Priority queues provide better service to HP data.. However, because each node makes the best decision locally, such a scheme may not be able to provide better service globally. Consider the case in which a node has an empty HP queue but a nonempty LP queue. This node will start injecting LP traffic into the network, which, due to the shared medium, may degrade the service provided to HP packets in nearby nodes. CAR and its enhancements, on the other hand, separate the traffic into two regions and hence eliminate most of the interference that can be caused by having both LP and HP traffic routed on the same paths.

Fig. 13 plots the fraction of HP data delivered to the sink. As the transmission range increases, the network becomes more congested, and more collisions occur. As a result, the performance of AODV degrades severely, and it routes less than 10 percent of HP data successfully. On the other hand, AODV+PQ and CAR-based schemes route a higher fraction of the data, although CAR-based schemes route more HP data than AODV+PQ for all ranges. At ranges larger than or equal to 130 m, CAR-based schemes route more than 90 percent of the data Finally, MCAR routes nearly all of the HP data, as it uses MAC-layer mechanisms to silence the conzone and its neighborhood in terms of LP traffic.

Fig. 14 shows the fraction of LP data routed successfully. Although our focus is to provide better service to HP data in the presence of congestion, CAR also effectively utilizes the uncongested off-conzone nodes to prevent severe degradation of LP data. Hence, in addition to improving HP delivery, CAR also enhances delivery of LP traffic as the range increases. The AODV delivery ratio decreases sharply as the range increases, while AODV+PQ routes the highest percentage of LP data. Note that since AODV+PQ routes less HP data and more LP data than CAR-based schemes, it is clear that priority-queue-based schemes alone are not sufficient to provide better service to critical data.

CAR routes more LP data than AODV as the range increases, since it prevents LP data from entering the conzone and getting dropped. AODV+PQ routes more LP data than CAR, because it does not as aggressively degrade service to LP data as CAR. At large ranges (i.e., in networks with few hops from the sink to the critical area) AODVþPQ routes more LP data and approximately the same amount of HP data as CAR. This is because in CAR, congestion may occur in off-conzone areas, as LP data from the conzone is routed out into such areas. MCAR drops virtually all LP data. This is due to the close proximity of the LP sinks to the HP source.

The applications described above and the MCAR combined MAC and routing algorithms with its modified RTS/CTS could be implemented in TinyOS . The environmental monitoring application has a timer that runs periodically. When the timer fires, the battery, light, and temperature sensors are polled, and the measurements are converted to digital values. If the temperature reading is above a user-defined threshold, which can be adjusted interactively by users via commands sent over the radio or USB backbone, the node transitions into the HP mode.

The primary challenge in implementing MCAR involved the strictly modular design of TinyOS. 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.

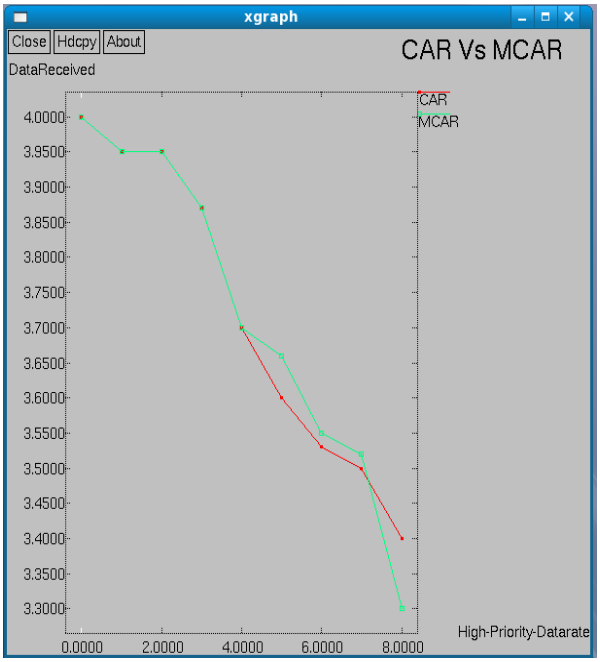

Fig.13. *Performance of High priority data*

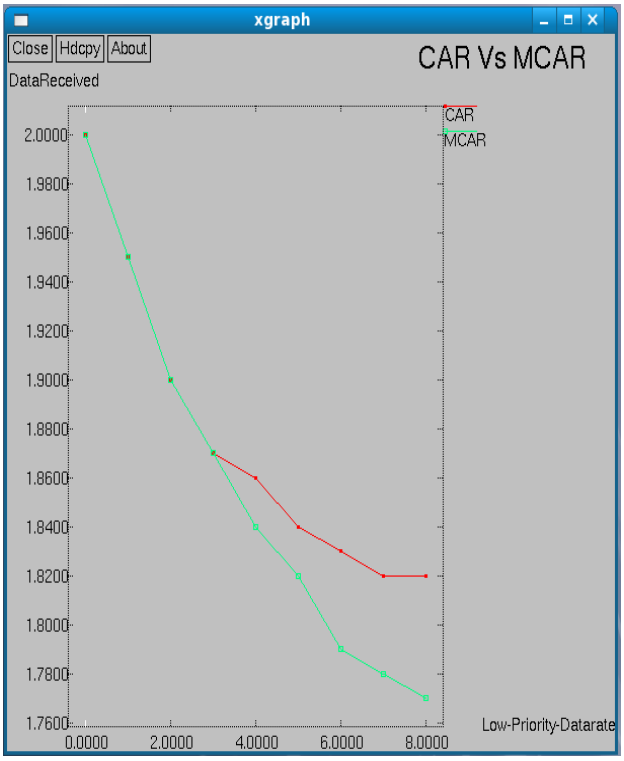

Fig.14. *Performance of low priority data*

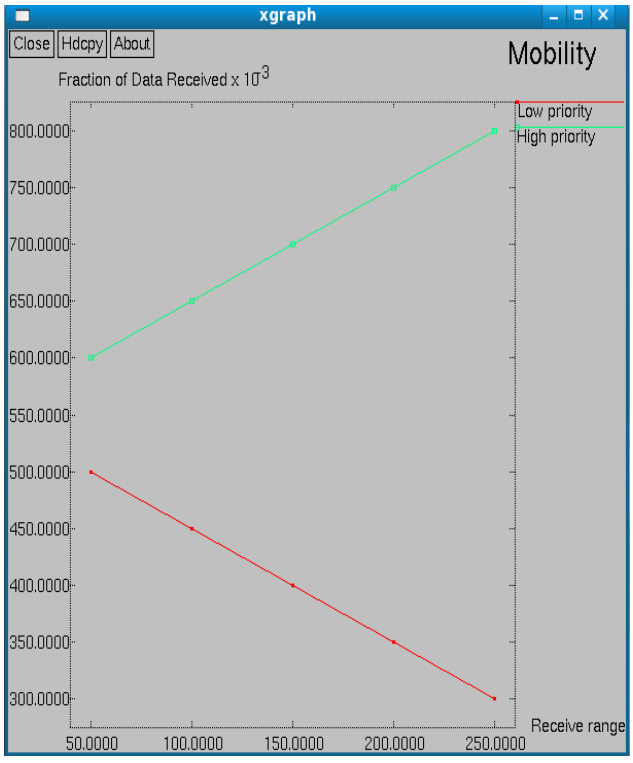

Fig.15. *Mobility*

5. Conclusion

Data delivery issues, in the presence of congestion in wireless sensor networks are analyzed. CAR, the differentiated routing protocol uses data prioritization. MCAR, which deals with mobility and dynamics in the sources of HP data.

CAR and its variants increase the fraction of HP data delivery and decrease delay and jitter for such delivery while using energy more uniformly in the deployment. CAR also routes an appreciable amount of LP data in the presence of congestion.

Additionally show that MCAR maintains HP data delivery rates in the presence of mobility and show that the route setup and tear-down times associated with the HP flows are minimal.

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 bursty HP traffic and/or mobile HP sources, MCAR is a better fit.

6. References

[1] Ahn, G. S. Campbell, A.T. F Hong, S.G. Miluzzo, E. (2006)"Funneling-MAC: A Localized, Sink-Oriented MAC for Boosting Fidelity in Sensor Networks,"

[2] Ahn, G.-S. Sun,L.-H. Versa. and Campbell,A.T.(2007) "Swan: Service Differentiation in Stateless Wireless Ad Hoc Networks," Proc.IEEE INFOCOM

[3] Akkaya, K. and Younis, M.F. (2006) "An Energy-Aware QoS Routing Protocol for Wireless Sensor Networks," Proc. 23rd IEEE Int'l Conf.

[4] Das, S.R .Perkins, C.E. and Belding-Royer, E.M. (2006) "Performance Comparison of Two on-Demand Routing Protocols for Ad HocNetworks," Proc. IEEE INFOCOM '00, pp. 3-12.

[5] Ee C.T and Bajcsy.R, (2005) "Congestion Control and Fairness for Many-to-One Routing in Sensor Networks," Proc. Second ACM Conf. Embedded Networked Sensor Systems (SenSys '04), pp. 148-161.

[6] Felemban, E. Lee, C.G. and Ekici, E. (2006) "MMSPEED: Multipath "Multi-SPEED Protocol for QoS Guarantee of Reliability.

N.Premalatha, M.E., Senior Lecturer Department of Computer Science and Engineering, Kongu Engineering College, Perundurai, India. She received the B.E. degree from Government College of Technology, Coimbatore, India in 1984, the M.E. degree from Kongu Engineering College, Perundurai in 2001 and she is currently pursuing the Ph.D. degree in Anna University Coimbatore, India. She has 9 years of experience in Teaching and 3 years of experience in industry. She received "Academic Excellence in Teaching award" from Kongu Engineering College. She is a member in IEEE. She had published 8 papers in National Conferences.

Dr.A.M.Natarajan, Chief Executive & Professor, Bannariamman Institute of Technology, Sathyamangalam. He received the B.E. degree from the P S G College of Technology, Coimbatore, India in 1968, the M.Sc. degree from the P S G College of Technology in 1970 and the Ph.D. degree from the P S G College of Technology, Coimbatore, India in 1994. He has 37 years of experience in Academic- Teaching, Research and Administration. He received "BEST ENGINEERING COLLEGE PRINCIPAL AWARD IN INDIA-2000" Instituted by Bharatiya Vidya Bhavan, Mumbai awarded by Indian Society for Technical Education , New Delhi. He also received "BEST OUTSTANDING FELLOW CORPORATE MEMBER OF COMPUTER ENGINEERING DIVISION-2001" from The Institute of Engineers (India),Coimbatore. And he was awarded "JAYA BEST PRINCIPAL AWARD-2007" by Jaya Engineering College, Thiruninravur for outstanding contribution to Technical Education over 37 years. He is the member in the Academic Council of the Anna University, Chennai. He is also a member in the Board of Studies, Anna University Chennai & Coimbatore. He is a member in ISTE, IEEE, IEI, ACM, CSI, CII etc. He had published 50 papers in National and International Journals and 70 papers in the National and International Conferences. He authored and published 10 Books.