Congestion and Position Aware Routing (CPAR) in Wireless Sensor Networks

Wenjun Gu, Prasun Sinha
Department of Computer Science and Engineering
Ohio State University
(gu,prasun)@cse.ohio-state.edu

ABSTRACT

We present a Congestion and Position Aware Routing protocl (CPAR) in sensor networks, which is based on Ad-hoc On-Demand Distance Vector (AODV) protocol [6]. We assume each node in the sensor network knows the positions of all its one hop neighbors and the sink, which can be achieved by GPS device or other mechanism that is mentioned below. Also we assume that each node knows the congestion information in the whole network, which can be achieved by a 3G channel. All the nodes in the sensor network will send congestion information that is piggybacked in the data packets or in a special congestion notification packet to the sink. The sink will collect these information, aggregate them and send them in the 3G signal to all the nodes. Using the position information of one hop neighbors and the sink, along with the global congestion information, each node can make a greedy decision in choosing one or a few of its one hop neighbors as forwarder(s) of its data stream.

Our simulation shows that the CPAR helps to decrease energy cost significantly while has comparable or better throughput than AODV in most cases.

Keywords

Wireless Sensor Network, Congestion and Position Aware Routing

1. INTRODUCTION

The development of wireless technologies and the wide potential application requirement make wireless sensor network (WSN) one of the hottest topics in the latest a few years. Wireless sensor network usually consists of a lot of sensor nodes ranging from dozens of to more than thousands of nodes and one or a few sinks. While sensor nodes are subject to a few constraints, including power, memory and processing ability, etc., we usually assume that the sinks are relatively free of these constraints. For example, it is much easier to recharge a few sinks than to recharge thousands of sensor nodes. In some applications, this is almost impossible since the sensor nodes are deployed in hostile atmosphere. Thus trying to save energy consumption while at the same time accomplishing the tasks becomes one of challenges in wireless sensor network.

Under most application scenarios, communication pattern in wireless sensor network is from sensor nodes to sink(s), which carries the data being collected and processed by the sensor nodes that is needed by the sink(s). Sometimes sinks need to sender data to sensor nodes, such as reprogramming or control tasks. Considering that wireless sensor network may span a wide area, it is not energy efficient to transmit data from all the sensor nodes to the sinks with one hop radio transmitter. So multihop transmission is a common way in wireless sensor network.

There has been much research done in wireless multihop network [1, 2, 3, 4, 5, 6], most of which are designed for wireless ad hoc network that is similar to wireless sensor network in some sense. Although wireless ad hoc network and wireless sensor network bear some similarities, such as energy constraint, mobility and multihop transmission, etc., there are a few major differences between them. First, all the nodes in wireless ad hoc network can be considered to be equal while sink(s) in wireless sensor network are different from the sensor nodes. Second, the sensor nodes in wireless sensor network are fixed or less prone to mobility compared with the nodes in wireless ad hoc network in some application scenarios. Third, the energy constraint is of more importance in wireless sensor network, in which the sensor nodes are unable to be recharged. These differences make most routing protocols designed for wireless ad hoc network unsuitable in wireless sensor network.

Congestion avoidance, detection and control is another hot topic in wireless network. Similar to the wired network, congestion can degrade the performance dramatically in terms of throughput, delay and delay jitter, etc. This problem becomes more serious in wireless sensor network since congestion costs the senders and forwarders much energy without getting the data to be delivered to the destinations. Thus efficiently detecting congestion and avoiding congestion is of more importance for wireless sensor network.

In this paper, we propose a Congestion and Position Aware Routing (CPAR) protocol for wireless sensor network. We assume all the sensor nodes in the sensor network know the positions of all their one hop neighbors and the sinks, which can be achieved by GPS device or other mechanism that is mentioned below. Also we assume that all the sensor nodes have the congestion information of the whole network, which can be achieved by a 3G channel. All the nodes in the sensor network will send congestion information that is piggybacked in the data packets or in a special congestion notification packet to the sinks. The sinks will collect these information, aggregate them and send them in the 3G signal to all the nodes. The traditional short range wireless radio channel is used by sensor nodes to send data, congestion information or position information to the sinks, while the 3G channel is used by the sinks to send aggregated and processed congestion information to all the sensor nodes in the network. These two channels do not collide with each other. Using the position information of one hop neighbors and the sinks, along with the global congestion information, each node can make a greedy decision in choosing one or a few of its one hop neighbors as forwarder(s) of its data stream. Our simulation shows that CPAR protocol helps to decrease energy cost while has comparable or better throughput than AODV in most cases.

The remainder of this paper is organized as follows. In section 2, we give an overview of the current wireless multihop routing protocols and wireless congestion control protocols. In section 3, we describe our Congestion and Position Aware Routing (CPAR) protocol in detail. Section 4 presents the results of our simulations. Section 5 describes some other ideas and our plan for future work and finally section 6 concludes the paper.

2. RELATED WORK

There has been much research on wireless multihop routing [1, 2, 3, 4, 5, 6], some of which are based on flooding while some others are based on geographic information. Also there have been some research on congestion control in wireless networks [7]. Here we give a brief overview.

2.1 Wireless multihop routing protocols

According to the routing mechanism and assumptions, we divide the current wireless routing protocols into two major classes. Destination Sequenced Distance Vector (DSDV) [5], Ad-hoc On-Demand Distance Vector (AODV) [6] and Dynamic Source Routing (DSR) [2] are representatives in the first class. These wireless routing protocols do not assume any geographic information, that means, the sender node does not know where all the other nodes are before communication. Data forwarding is based on routing tables that is set up either a prior or on demand. Basically, flooding is used in finding the routing path. DSDV requires all the nodes to exchange routing information on a periodic basis, which helps to set up the routing tables before communication. AODV and DSR use on demand scheme, in which routing table is set up whenever it is needed. The major differences between AODV and DSR is that AODV forwards the data in a distributed way, while DSR does it in a centralized way, in which the sender itself has full control of the routing path.

Although AODV is designed for wireless ad hoc network, it can be used in wireless sensor network directly with the destination be the sinks. The advantage of AODV to DSDV is that routing paths are set up whenever necessary, which helps to decrease the storage overhead for routing tables and the routing overhead in building the routing paths. The storage overhead is decreased because the total number of paths needed to be set up is the same as the number of sensor nodes that need to send data to the sinks instead of the number of all the sensor nodes as in DSDV. The routing overhead is decreased because in AODV, periodic routing information exchange is not needed. However, the disadvantage of AODV compared with DSDV is that there will be some delay between when the sender node sends routing request and when the routing path is built.

Compared with DSR, AODV achieves better scalability since it is totally distributed. The routing path is maintained by the sender node and all the intermediate forwarders along the path. The sender only knows the next hop node on the path and all the intermediate forwarders only knows their next hop nodes. But in DSR, the routing path is maintained by the sender node itself. The sender nodes need to put the path information in the packet such that each intermediate forwarder can find which node is the next hop forwarder. AODV saves packet size since no path information is needed in the packet, while it needs more storage overhead to store all the paths in all the intermediate forwarders' routing tables.

Our CPAR protocol is based on AODV because it saves routing overhead and storage overhead compared with DSDV and is more scalable and saves transmission overhead compared with DSR. We try to decrease routing overhead further based on geographic information and congestion information that will be described in the next section.

2.2 Geographic position aware protocols

Greedy Perimeter Stateless Routing (GPSR) [3], Location Aided Routing (LAR) [4] and Distance Routing Effect Algorithm for Mobility (DREAM) [1] are representatives for the second class of wireless multihop routing protocols. These protocols assume that the sender nodes have some geographic information of the destination nodes, which can help to identify the direction of the routing path efficiently while at the same time save energy on routing packets compared with flooding schemes used in the first class of protocols.

GPSR assumes that each sensor node knows the geographic information of the sinks and all its one hop neighbors. The sender and all the intermediate nodes always chooses one of its neighbors that is the closest to the sink, which is called greedy forwarding phase. When the sender or the intermediate node itself is closer than all its one hop neighbors, it turns into perimeter phase until reaching a node that is closer than the node when the perimeter phase begins. In this way, routing overhead can be decreased dramatically, especially for dense network.

LAR is designed for mobile networks. It assumes that each node knows the geographic information and some speed information of all other nodes, which may not be up to date. Whenever a sender node needs to communicate with a destination node, it computes an expected area that contains the destination node with high probability based on the geographic information, the speed information of the destination node and the time elapsed since the last time when these information were obtained. After that, the sender node computes the request zone based on its current position and the expected zone of the destination node computed above. Only the nodes in the request zone need to forward the routing request. In this way, routing overhead can also be decreased significantly.

Similar to LAR, DREAM is also designed for mobile networks. It assumes that each node knows the geographic information and some speed information of all other nodes, which may not be up to date. Whenever a sender node needs to communicate with a destination node, it also computes an expected area such that the probability that this area contains the destination node with at least the probability p that is given based on the geographic information, the speed information of the destination node and the time elapsed since the last time when these information were obtained. Routing overhead can also be decreased in this way.

Our CPAR protocol bears some similarities with the three protocols above. Similar with GPSR, we also assumes that each sensor node knows the geographic information of the sinks and all its one hop neighbors. Different from GPSR, we assume that each nodes has the global congestion information that is obtained by our special 3G channel. Also, we make the number of next hop nodes to be a parameter, which is always 1 in GPSR. Besides that, we use a different strategy in choosing the next hop neighbor(s), which is based on minimum degree instead of minimum distance. We believe that this is better in achieving energy efficiency, which will be described in the next section.

Similar to LAR and DREAM, our CPAR also requires that each sender node or forwarder node needs to compute a request area. Our request area is based on minimum degree, which is different from LAR's rectangle request area and minimum distance request area. Another major difference is that we assume the nodes are fixed, so we do not need any speed information and we guarantee that the destination is in the request area. One other difference is that we utilize global congestion information to achieve smart request area computation.

2.3 Wireless congestion control protocol

There has also some research on congestion detection, avoidance and control in wireless network. Congestion Detection and Avoidance (CODA) [7] is one of the most important work. In CODA, congestion is detected by the buffer size and the channel load. Congestion control is achieved by an open loop and a closed loop control mechanism, both of which are similar with the corresponding schemes used in wired network. In our CPAR protocol, we also compute the congestion state based on the buffer size and the channel load. The difference is that we compute the congestion information for each node no matter whether it is congested or not. We believe that more congestion information will not only help to increase throughput and decrease energy cost by avoiding potential congestion, but also help to balance the load in the whole network. The mechanism to extract congestion information is left for future work. In this paper, we assume that congestion information is always available in an up to date manner.

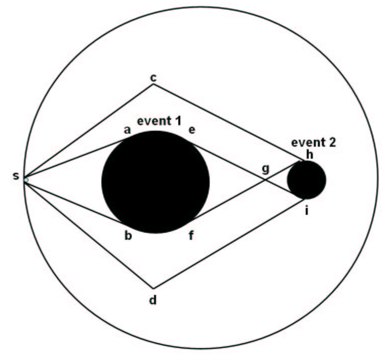

Figure 1: Request Area in CPAR Protocol

3. CONGESTION AND POSITION AWARE ROUTING (CPAR) PROTOCOL

3.1 Assumptions

We assume that each node in the sensor network knows the positions of all its one hop neighbors and the sinks, which can be achieved by GPS device or hello messages between each node and its neighbors. This mechanism is enable in AODV. Since sinks are usually immobile nodes and have limited number, it is relatively easy for each sensor node to obtain these information. When the whole network is immobile, the hello messages are used to get the neighbor's positions at the first time and are used to make sure that those neighbors are alive later on. The mechanism of obtaining these information is out of the scope of this paper.

We also assume that each node knows the congestion information in the whole network, which can be achieved by a 3G channel in our CPAR protocol. All the nodes in the sensor network will send congestion information that is piggybacked in the data packets or in a special congestion notification packet to the sink on a periodic basis. The sink will collect these information, aggregate them and send them in the 3G signal to all the nodes. The 3G channel does not interfere with the ordinary channel used for data transmission. In order to achieve energy efficiency and computation efficiency, congestion information computation, transmission, aggregation and broadcasting should be done efficiently, which is out of the scope of this paper.

3.2 Protocol details

Our CPAR protocol focuses on computing the request area and routing packets forwarding based on the congestion and position information we mentioned above. Our goal is to achieve energy efficiency by decreasing the number of routing packets with minor or no negative effect on the performance in terms of total throughput and the throughput of the event behind the congested area.

We observe that AODV uses flooding to get the routing path, which will consume too much energy in a wide and dense sensor network. In figure 1, there are two events in the sensor network that is a circle for simplicity. There is only one sink in the network, which is denoted by s. Let us assume that event 1 happens first and the area surrounded by node s, a, e, f and b are a little congested. When event

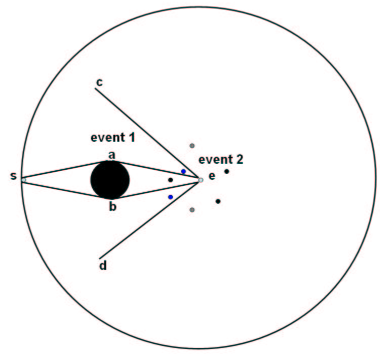

Figure 2: Next Hop Forwarder Selection in CPAR Protocol

2 happens, according to AODV, it will send routing request in a flooding way, thus almost all the nodes in the network will receive and retransmit the routing packets, which is unnecessary. We claim that the routing request should be sent and forwarded in only part of the network, which we call request area. In figure 1, there are two request areas for event 2. The first is surrounded by node s, c, h, i, e and a. The second is surrounded by node s, d, i, h, f and b. Since the area surrounded by node s, a, e, f and b is congested, it is unwise to send and forward routing request packets to this area since the routing request packets and the routing reply packets are prone to getting lost, which will only waste some energy. Even if the routing request packets and routing reply packets do not get lost, sending data into this area will increase the congestion state and degrade the performance.

As to the area surrounded by node e, f and g, it may also be wise not to send routing request packets into this area since it is close to event 1. Routing packets and data in this area are prone to collide with the traffic generated by event 1. As to the area out of our request areas, it is not necessary to send routing packets into this area if the network is dense enough. This is because the request area has enough nodes to be potential forwarders while at the same time it is not congested. Sending routing packets out of the request area will just waste some energy and sending data packets out of the request area will be prone to worse performance since the data follows a longer route.

Now the problem turns to be how to compute the request area. To be more exact, the problem becomes how to choose the next hop nodes for routing packets. In figure 2, we give a simple example to help clarify our algorithm in choosing the next hop nodes. Similar to figure 1, s is the only sink in this circular sensor network and event 1 happens before event 2. We assume the the total rate of event 1 is high enough such that the area surrounded by node s, a and b is a little congested. We let event 2 to contain only one node e to simplify the clarification. Suppose now event 2 happens. According to the global congestion information, node e will not choose its neighbors in the area surrounded by node a, b and e. After that, node e will computer the degree between two lines, one of which is between node e itself and one of its neighbors and the other is either line a-e or line b-e depending on the position of the neighbor node. Node e will choose one or a few of its neighbors that have the smallest degree. In our CPAR protocol, the number of neighbors chosen is a parameter that can be adjusted according to the network condition. If this number is 2, node e will choose 2 of its neighbors as the forwarders and put their node IDs in the routing request packet.

Whenever all of node e's neighbors receive the routing request packet, they will check their node ID with the node IDs in the packet. If its node ID is in the packet, that means it is chosen as the forwarder. This node should computes its next hop forwarders in a similar way as above based on the congestion information and position information. Otherwise, the node will just drop the packet since it is not chosen as the forwarder. This procedure proceeds until a node that has already have the route to the sink or the sink itself. In this way, only a limited number of nodes in the sensor network need to broadcast the routing request packet for event 2, which decreases the routing overhead in a significant way.

In our implementation, we introduce two modification. First, node b is actually below the lowest node in event 1. The distance is the transmission range of a sensor node. We introduce this modification because we hope the rerouting path will be less likely to collide will some existing paths built by event 1. Similarly, node a is above the highest node in event 1 with a distance around the transmission range. Second, node e will not choose its neighbor as the next hop forwarder if it is in the opposite direction of the sink. In figure 2, node e will not choose the right most two nodes no matter whether they are chosen according to the above procedure. In a sparse network, the node may not be able to find enough neighbors as the forwarders. Whenever this happens, the node will choose all its neighbors as the next hop forwarders since in this case the node is in a similar situation as that in GPSR when the protocol state transfers from greedy phase to perimeter phase. This can help improve the reliability of our protocol.

3.3 Preliminary analysis

In this subsection, we present the result of our preliminary analysis on the potential decrease in routing packets our CPAR protocol can achieve. We assume the sensor network is a circle and there is only one sink s in the network that is located on the edge of the circle. The first event is between sink s and the second event, both of which are all small circles. Node s, event 1 and event 2 are in a line and event 2 is located in the middle of the network. Generally, the analysis should consider many factors, including the network area, the number of sinks and their positions, the number of events and their positions, the density and evenness of the sensor nodes, the number of neighbors chosen as the next hop forwarders, etc. We make some assumptions above in order to simplify the analysis and we leave the general analysis as the future work.

Each node in event 2 will choose its next hop forwarders using the above mentioned algorithm. If we assume the radius the network is R, the transmission range of each sensor node is 250 meters, the total number of sensor nodes in the network is N, the number of neighbors chosen as the next hop

Table 1: Simulation Parameters in Scenario 1

Sink Position (meter)	(0,1500)
Time duration (sec)	0-10
number of nodes in small event	1
its position (meter)	(2265,1467)
number of nodes in big event	3
their positions: node 1 (meter)	(1241,1650)
node 2 (meter)	(1358,1415)
node 3 (meter)	(1128,1096)
high data rate (Kbps)	88
low data rate (Kbps)	44

Table 2: Simulation Parameters in Scenario 2

Sink Position (meter)	(0,1500)
number of nodes in big event	3
Time duration (sec)	0-10
their positions: node 1 (meter)	(1241,1650)
node 2 (meter)	(1358,1415)
node 3 (meter)	(1128,1096)
number of nodes in small event	1
Time duration (sec)	3-8
its position (meter)	(2265,1467)
high data rate of event 1 (Kbps)	88
low data rate of event 1 (Kbps)	44
high data rate of event 2 (Kbps)	40
low data rate of event 2 (Kbps)	20

forwarders is n and the sensor nodes is evenly distributed in the network, we can get the average number of neighbors each node has as follows.

$$Ne = N * 250^2/R^2 - 1 \quad (1)$$

If we assume the request area in figure 1 is surrounded by two arcs, we can get the performance improvement as follows.

$$PI = 1 - t * n/(Ne * R) \quad (2)$$

Here, t is the length of the request area, which depends on the sizes and positions of the two events. On average t is about 1.5. If you let the network density to be 500 nodes in 9 square kilometers and let n be 2. PI will be 0.68 that means that 68 percent of the routing packets can be saved, which is quite significant.

4. SIMULATIONS AND RESULTS

4.1 Simulation Environment

In our simulation, we use a 3km * 3km square as the network size, the number of sensor nodes in 500 and the number of sink is 1. The sensor nodes are deployed randomly using ns-2's setdest utility tool. The sink is located on the edge of the network. Considering the space limit of the paper, we present the result of two network scenarios here. The first one consists of one event and the second one consists of two events. In the first scenario, we tried both big event and small event, high data rate and low data rate. In the second scenario, we tried different combination of the data rates of the two events. We summarize the parameters of our simulation in table 1 and 2.

The metrics we use are number of sent AODV routing packets, number of received AODV routing packets, delivery ratio of one event and total delivery ratio of two events. The AODV routing packets include both routing request packets and routing reply packets.

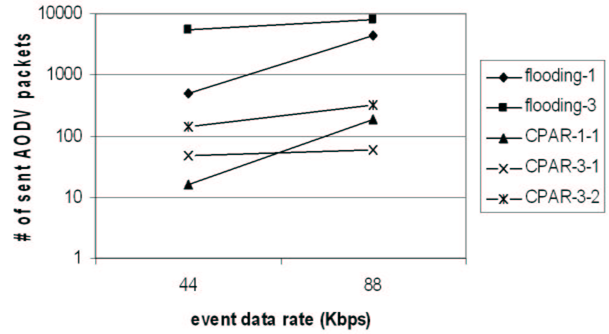

Figure 3: Number of Sent AODV Packets in Scenario 1

4.2 Results and Explanation

Figure 3 shows the number of sent AODV routing packets in the first simulation scenario. There are five curves in the graph, in which flooding-1 means traditional AODV protocol with small event, flooding-3 means AODV with big event, CPAR-1-1 means our CPAR protocol with small event and only one neighbor is chosen as the next hop node, CPAR-3-1 means CPAR with big event and one neighbor chosen as the next hop node and CPAR-3-2 means CPAR with big event and two neighbors chosen as the next hop node.

In figure 3, it shows that the number of sent AODV routing packets in flooding-1 and flooding-3 is much more than that in three CPAR cases. This is because CPAR does restricted flooding. It is obvious that choosing two neighbors will result in more routing packets than choosing one neighbor. For both AODV and CPAR, the number of sent AODV routing packets increase either when the event data rate increases or when the event is big. It is obvious that the number of sent AODV routing packets increases with the data rate since the routing packets and data packets are subject to loss in higher data rate case, which triggers more routing packets to be sent. For big event, this increase is less significant since there has already more collision among different flows in big event already. The number of sent routing packets is more in big event case than that in small event case since different flows among big event are prone to collision, which results in packets loss and routing packets retransmission.

The improvement is significant since each AODV packet is 48 bytes while the data packet is 220 bytes in our simulation. Let us compare flooding-1 and CPAR-1-1 in high data rate case. For flooding-1, the routing packets stand for about $4531 * 48 * 8/(4531 * 48 * 8 + 880000) = 0.66$ of all the traffic in the network, that means, two thirds of the traffic are not data packets. While in CPAR-1-1, the percentage of routing packets is about $191 * 48 * 8/(191 * 48 * 8 + 880000) = 0.077$, which is a significant decrease. According to our simulation, CPAR performs much better in high data rate, big and dense

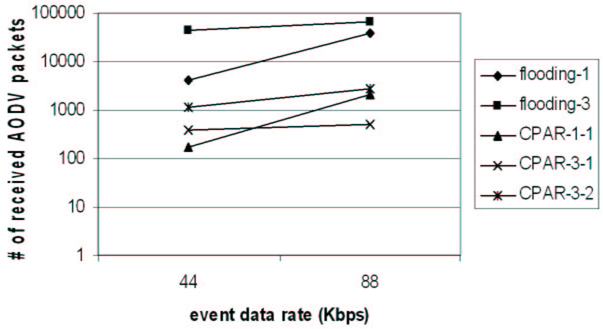

Figure 4: Number of Received AODV Packets in Scenario 1

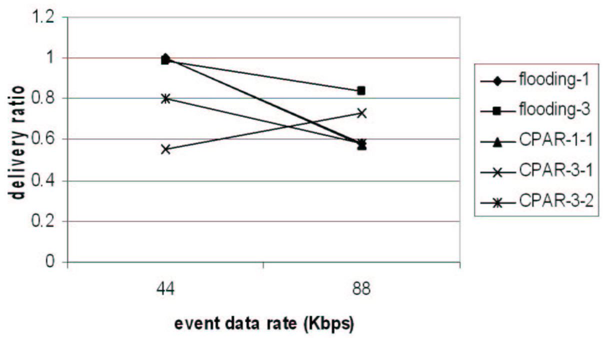

Figure 5: Delivery Ratio in Scenario 1

events cases, which suffers from routing packets explosion. The improvement is even better than our analysis in the previous section since choosing one neighbor is an extreme case for our analysis. Our analysis will be close to the data we get from simulation when the number of neighbors chosen increases.

Figure 4 shows the number of received AODV routing packets in the first simulation scenario. The curves are similar to those in figure 3 except that the number of received routing packets is about 8 to 10 times that of sent routing packets. This is in our expectation because the average number of neighbors a node has in our scenario is between 9 and 10 according to our analysis above.

Figure 5 shows the delivery ratio of the data packets in the first simulation scenario. It is obvious that the delivery ratio decreases when the data rate is high since there is more congestion in the network. Given the same data rate, a big event has higher delivery ratio since the data are delivered in multiple paths, which smoothes the congestion state in the network. Generally the delivery ratio in our CPAR with two neighbors chosen is slightly worse than or similar to that in AODV, which is better than that when one neighbor is chosen. This is because choosing fewer neighbors as forwarders sometimes results in the fact that all the routing packets are lost in the way. If we choose too few neighbor, the routing path requesting may not be stable, while if we choose too

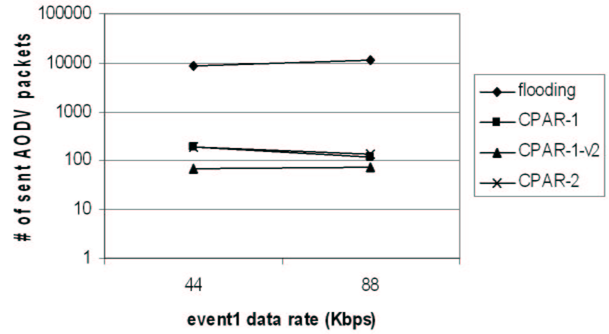

Figure 6: Number of Sent AODV Packets in Scenario 2 (event 2 rate is 20Kbps)

many neighbors, it suffers from routing packets explosion. The optimal number of neighbors chosen depends on the congestion state in the network, the density of the network and the application requirement. For example, in less congested area, choosing smaller number is more efficient while in quite congested situation, choosing more neighbors will be more reliable. For application that requires reliable delivery, more neighbors should be chosen, while for applications that requires energy efficiency, fewer neighbors should be chosen. In our simulation, choosing two neighbors is good enough in terms of throughput in most cases since the network is dense.

We observe that there are some abnormal data we obtain in our simulation. In figure 5, the delivery ratio of CPAR-3-1 is lower in low data rate case than that in high data rate case. We studied the trace file and found that it is because in the former case, data transmission happens to be blocked some time during the simulation. The packets are dropped until the end of the simulation instead of immediately after the packet is received, which results in the absence of routing request retransmission and the delivery ratio becomes zero from then on. This also shows that choosing only one neighbor is not reliable since the path is blocked when only one nodes on the path gets congested or blocked. The other abnormal data we obtained is the delivery ratio of CPAR-3-2 in high data rate, which is lower than that of CPAR-3-1. After studying the trace file, we found that this follows the same reason as that above. This means that selecting two neighbors is generally reliable, but occasionally it happens that routing packets gets blocked on the way. This abnormal situation happens in some simulations that we will mention later. We suspect this is related with the current implementation of AODV in the current version of ns-2.

Figure 6 and 7 show the number of sent AODV routing packets with different combination of the data rates of the two events in the second simulation scenario. In figure 6 the data rate of event 2 is 20Kbps while it is 40Kbps in figure 7. There are four curves in each graph, in which flooding means the traditional AODV scheme, CPAR-1 means our CPAR protocol with only one neighbor chosen as the forwarder, CPAR-2 means CPAR with two neighbors chosen as forwarders and CPAR-1-v2 is a modification that will we

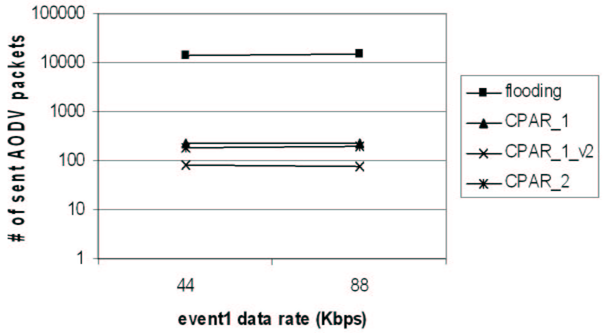

Figure 7: Number of Sent AODV Packets in Scenario 2 (event 2 rate is 40Kbps)

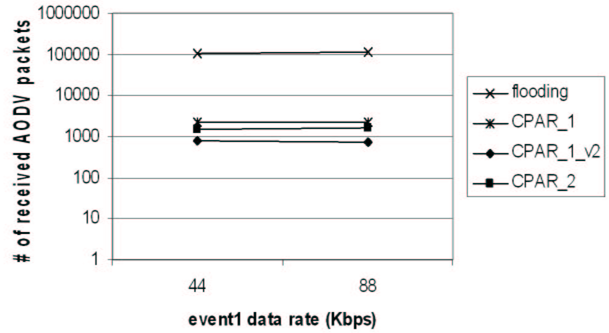

Figure 9: Number of Received AODV Packets in Scenario 2 (event 2 rate is 40Kbps)

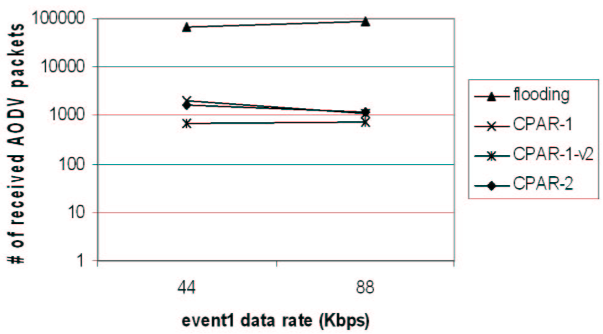

Figure 8: Number of Received AODV Packets in Scenario 2 (event 2 rate is 20Kbps)

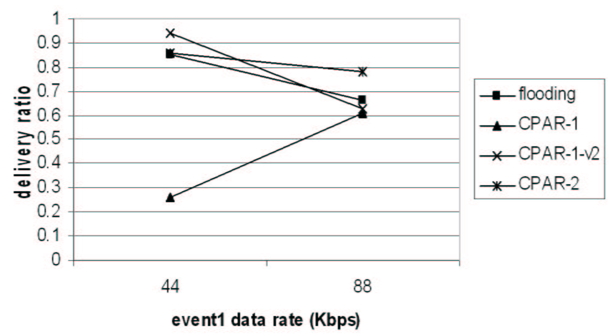

Figure 10: Total throughput in Scenario 2 (event 2 rate is 20Kbps)

will describe below. In CPAR-1, each source node in event 1 chooses one neighbor and the chosen neighbors choose the next hop neighbor independently. It is possible that one node will be chosen as the forwarder by multiple flows. We will show later that this is not stable in some simulation cases, which results in degraded performance. This is why we introduce CPAR-1-v2, which is a modification of our original CPAR-1. In CPAR-1-v2, the source nodes and all the forwarders for event 1 collaborate in some way such that no node will be chosen as the forwarder by multiple flows. This can be achieved in a similar way as to the sharing of the congestion state by the 3G signal. In our simulation, we make this a global variable.

We observe in figure 6 and 7 that the number of sent routing packets in all CPAR schemes is much fewer than that in AODV scheme, which are the same as those in figure 3 and in our expectation. The number of routing packets increases slightly with the data rate because the packets are prone to loss and will more likely to trigger routing packets retransmission in high data rate situation. Couple abnormal data comes in figure 6 for both CPAR-1 and CPAR-2 cases, which has slightly fewer sent routing packets when data rate increases. This is brought by the same reason we mentioned above. Figure 8 and 9 show the number of received AODV routing packets with different combination of the data rates of the two events in the second simulation scenario. The pattern is similar to figure 6 and 7 except a difference of a factor between 8 and 10, which is expected from our analysis.

Figure 10 give the total delivery ratio of two events and figure 11 gives the delivery ratio of event 2 that is behind the congested area in the second simulation scenario. We only show the result when the data rate of event 2 is low because of space limitation. We observe that CPAR-1-v2 and CPAR-2 achieve close or even better performance than AODV since CPAR tries to avoid congested area for new event such that the total throughput can be increases a little in some cases. This improvement is limited by the location of the sink and the topology of the area around the sink. In our network topology, the upper limit of the throughput for the sink is around 90Kbps. So we expect our CPAR-2 to achieve better performance when the sink is located in the middle of the network. There are some abnormal data for CPAR-1 case, in which the delivery ratio is abnormally low when the data rate of event 1 is low. This is because of the same reason we mentioned above.

To sum up, our CPAR protocol decrease routing packets significantly while being reliable and can achieve comparative performance as AODV at most cases. Sometimes our

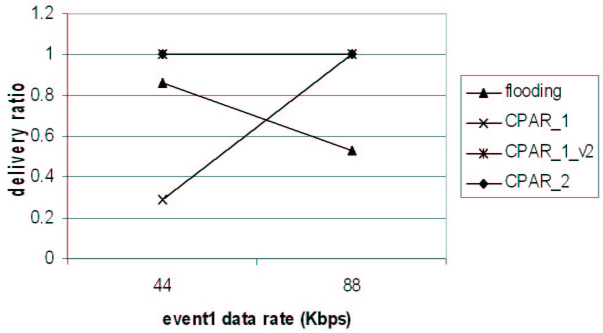

Figure 11: Event2 delivery ratio in Scenario 2 (event 2 rate is 20Kbps)

CPAR protocol can achieve better performance because of congestion avoidance, while sometime the performance is abnormally degraded because of one reason we point out above.

5. DISCUSSION AND FUTURE WORK

5.1 Discussion

In our CPAR protocol, we select the next hop neighbors based on minimum degree, which is different from minimum distance used by GPSR. The advantage of minimum degree based scheme is that the routing path is close to a straight line, which is ideal for energy efficiency when the energy used in transmitting and receiving packets is proportional to the length of the radio link. The disadvantage of minimum degree based scheme is that the routing path chosen may be of more hops than the minimum distance based scheme. We can prove that the probability that the neighbor chosen by minimum degree based scheme is the same as the neighbor chosen by minimum distance based scheme is quite high, especially in dense networks.

The reason why we compute the request area in our implementation of CPAR protocol is that we hope in this way the probability that the rerouting path collides with the current paths is minimized while at the same time as close to straight line as possible. Actually we can modify the way we compute the request area in a few ways. First, the request area can include the area between the closer event and the further event since this area is not congested by the closer event. In order to prevent collision, we can exclude part of this area that is close to the closer event from the request area. We believe the network bandwidth resource can be used more efficiently by this modification. Second, the distance between request area and the closer event may not be necessary one transmission range. It could depend on several factors, such as the density of the network, the congestion state of the closer event and the area around it, the data rates of these two events, the position of the events in the network and how possible the around area will be used by future events, etc. Third, when the area around the sink is highly congested, we may want the request area to be as far away from the area between sink and the closer event as possible, which is easier to achieve when the sink is not deployed on the edge of the network.

In this paper, we only do analysis and simulation on simple two events scenario. For general scenario, it may not be efficient to broadcast all the detailed congestion information frequently. We can divide the whole network into small grids and aggregate the congestion information for each grid. Rerouting path can be selected by some Quality-of-Service routing protocols that have been brought forward for wired network. The metric we can use here is the congestion state and hop count. Since these two metrics collide with each other, we need to set different weight for these two metrics based on the application requirement. We can also broadcast the congestion information in a less frequent and less global way. In the former case, the congestion information may not be accurate. Imprecise state routing protocols may be of help here. In the latter case, we may save energy by not broadcasting the congestion information of less congested area.

5.2 Future work

The following is our plan for the future work:

- Compute congestion information from MAC layer or physical layer in a periodic way, which may be similar as the RTT computation used in TCP.
- Send congestion information to sinks efficiently. If may not be good for all the forwarders to put their congestion information in one packet since the loss of this packet results in a lot of congestion information losses. Probabilistic piggybacking may be suitable.
- Aggregate congestion information efficiently and collaborate with other sinks. The aggregation can be based on either time scale or geographic division.
- General routing algorithm for general network scenarios. When there are multiple events, multiple congested areas and multiple sinks, a general routing protocol that is similar to the QoS routing protocols in wired network is necessary, which may also need to tolerate some degree of imprecise congestion information. In this case, a general next hop neighbor selection criteria may be based on a few factors that are mentioned in the previous subsection.
- For mobile network, we need speed information to be used in request area computation. Schemes designed in LAR and DREAM may be useful here. Considering the variation of the speed, our protocol should tolerate some degree of imprecise speed information.
- Nodes in sensor network may become dead either because of no power or some other physical reasons. Periodic hello messages are useful here. In some cases, the sinks need to request the sensor nodes to make sure that they are alive.
- We also plan to get more data from more diverse network scenarios and wish to analyze the effect of some parameters on the performance and energy cost in our protocol.

6. CONCLUSIONS

In this paper, we presented a congestion and position aware routing protocol (CPAR) based on local geographic information and global congestion information for wireless sensor network. We use AODV as the underlying routing mechanism. The local position information is achieved by GPS device or periodic hello messages. The congestion information is computed by each node on a periodic basis, transmitted to the sinks piggybacked on data packets or in specific congestion state packets, aggregated by the sinks and broadcast on the 3G channel.

By pure mathematical analysis and ns-2 simulation, we show that our CPAR protocol can achieve energy efficiency by decreasing routing packets significantly, especially in dense network, while at the same time improve throughput and have comparable or better performance than that using traditional AODV protocol. We believe that CPAR is especially useful for those wireless application in which energy efficiency and prolonging network life is important.

7. ACKNOWLEDGMENTS

The author would like to thank Sriram Chellappan for the helpful discussion.

8. REFERENCES

[1] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward. A distance routing effect algorithm for mobility (dream). *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 76–84, 1998.

[2] D. B. Johnson, D. A. Maltz, and J. Broch. *Ad Hoc Networking*. Addison-Wesley, Boston, 2001.

[3] B. Karp and H. T. Kung. Greedy perimeter stateless routing for wireless networks. *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000)*, pages 243–254, August 2000.

[4] Y.-B. Ko and N. H. Vaidya. Location-aided routing (lar) in mobile ad hoc networks. *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 66–75, 1998.

[5] C. E. Perkins and P. Bhagwat. Routing over multihop wireless network of mobile computers. *SIGCOMM'94: Computer Communications Review*, 24(4):234–244, October 1994.

[6] C. E. Perkins and E. M. Royer. Ad hoc on-demand distance vector routing. *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, February 1999.

[7] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell. Coda: Congestion detection and avoidance in sensor networks. *ACM SenSys'03*, October 2003.