Sink Repositioning For Enhanced Performance in Wireless Sensor Networks

Kemal Akkaya, Mohamed Younis and Meenakshi Bangad
Dept. of Computer Science and Electrical Engineering
University of Maryland Baltimore County
Baltimore, MD 21250
{kemal1, younis, bangadm1}@cs.umbc.edu

Abstract. Most of the energy aware routing approaches for unattended wireless sensor networks pursue multi-hop paths in order to minimize the total transmission power. Since almost in all sensor networks data are routed towards a single sink (gateway), hops close to that sink become heavily involved in packet forwarding and thus their batteries get depleted rather quickly. In addition, the interest in optimizing the transmission energy tends to increase the levels of packet relaying and thus makes queuing delay an issue, especially for real-time traffic. In this paper we investigate the potential of gateway repositioning for enhanced network performance in terms of energy, delay and throughput. We address issues related to when should the gateway be relocated, where it would be moved to and how to handle its motion without negative effect on data traffic. We present two approaches that factor in the traffic pattern for determining a new location of the gateway for optimized communication energy and timeliness respectively. The gateway movement is carefully managed in order to avoid packet losses. The proposed approaches are validated in a simulated environment.

1. Introduction

Networking unattended wireless sensors is expected to have significant impact on the efficiency of many civil and military applications, such as disaster management, environment monitoring, combat field surveillance and security [1][2][3][4][5][6]. These systems process data gathered from multiple sensors to monitor events in an area of interest. For example in a disaster management setup large number of sensors can be dropped by a helicopter. Networking these sensors can assist rescue operations by locating survivors and assessing their medical conditions, identifying risky areas and making the rescue crew more aware of the overall situation. Such application of sensor networks not only can increase the efficiency of rescue operations but also ensure the safety of the rescue crew. In addition, the use of networked set of sensors can facilitate monitoring of forests for an early detection of fires and limit personal involvement in risky military reconnaissance missions. Security applications of sensor networks include intrusion detection and criminal hunting.

Sensors measure ambient conditions related to the surrounding environment and send the collected data, usually via a radio transmitter, to a command center either directly or through a gateway. Sensors used in many of the emerging applications are disposable and expected to last until their energy drains. Therefore, energy is a very scarce resource for such sensor systems and has to be managed wisely in order to extend the life of the sensors for the duration of a particular mission. Energy efficient data routing in wireless networks generally pursues multi-hop paths in order to minimize the total transmission power. Transmission power of radio circuits is generally proportional to distance squared or even higher order for environments rich with obstacles and interference sources. The basic idea of multi-hop paths is to shorten the distance so that significant power savings can be achieved. Therefore, the bulk of research on routing in wireless sensor networks mostly aims at maximizing the lifetime of the network, allowing scalability for large number of sensor nodes and supporting tolerance of sensor's damage and battery exhaustion [7][8][9][10][11][12].

The time for the first node to die is one of the most popular performance metrics for sensor networks. When it takes long for the first sensor to run out of energy, the network topology stays stable and data routing is performed efficiently. Since in almost all sensor networks data are routed towards a single sink, the gateway in our model, hops close to that sink become heavily involved in packet forwarding and thus their energy supply gets depleted rather quickly. When the gateway is stationary, hops that are further from the gateway have to be picked to substitute those close sensors, which are out of energy. Such scenario increases the total transmission power and gradually limits sensor coverage in the network and eventually makes the network useless. If the gateway has limited motion capability it will be desirable to relocate the gateway close to an area of heavy traffic or near loaded nodes in order to decrease the total transmission power and extend the life of nodes on the path of heavy traffic.

Gateway relocation can also be beneficial in applications involving real-time traffic. In such applications, data paths are carefully established so that certain end-to-end delay requirements are met [13][14]. The quality of service achieved in these applications can start to diminish with increased volume of real-time data and most of the packets can miss their specified deadlines. In order to enhance timeliness in such situations, one of the solutions is to explore gateway's ability to move to a location where the volume of real-time data is high. Gateway relocation in those circumstances can balance the traffic load among multiple nodes and hence decrease the miss rate of real-time packets. Examples of scenarios where relocation would be feasible are when the gateway is a laptop computer or other portable devices on the backpack of a rescue crew who is not expected to travel long distances.

Relocating the gateway during regular network operation is very challenging. The basic issues are when it would make sense for the gateway to be relocated, where the gateway should go and how the data traffic will be handled during the gateway's movement. The relocation of the gateway first has to be motivated by inefficient pattern of energy depletion or a non-tolerable increase in the number of missed deadlines when real-time packets are involved, even if it is the best possible network operation given the traffic distribution and network state at that time. Once the gateway detects such conditions, it should identify its most suitable location in order to enhance network performance. It is worth noting that in some setups it may also be desirable to continually adjust the gateway's location to better serve the application, even if its current position is not causing concerns.

Finding an optimal location for the gateway in a multi-hop network is very complex problem, NP hard in nature. The complexity is mainly resulting from two factors. The first factor is the potentially infinite possible positions that the gateway can be moved to. Second, for every interim solution considered during the search for an optimal location, a new multi-hop network topology needs to be established in order to qualify that interim solution in comparison to the current or previously picked location in the search. A mathematical formulation of the problem would involve huge number of parameters including the positions of all deployed sensors, their state information such as energy level, transmission range, etc., and the sources of data in the networks. Given the large number of nodes typically involved in applications of wireless unattended sensor networks, the pursuance of exhaustive search will be impractical. In addition, the dynamic nature of the network makes the sensor state and sources of data variant and the optimization process would have to be repeated when the gateway is to be moved multiple times.

Besides such complexity of the optimization problem, the gateway has to perform a trade-off analysis between the gain achieved by going to a new location and the overhead that the relocation imposes on sensors. For example, when energy metrics are of utmost concern the impact on the lifetime of an individual sensor has to be considered as well as system level metrics such as average energy per packet. If the relocation is justified, the gateway moves towards that location. While the gateway is on the move, it must ensure that data are not lost. Data loss can be caused by packet drops when the gateway moves out of the transmission range of some of the sensors that directly communicate with the gateway.

In this paper, we study the three issues of when, where and how to relocate the gateway node. We pursue a heuristic search that is based on the current network topology and traffic pattern. We first consider relocating the gateway under unconstrained traffic. We use the traffic density times the transmission power as a metric for monitoring the network operation and searching for the best gateway location. Our approach tracks the changes in the nodes that act as the closest hop to the gateway and the traffic density going through these hops. If the distance between the gateway and some of the nodes that are in direct communication is larger than a threshold value, the gateway will qualify the impact of these nodes on the overall network lifetime by considering the number of packets routed through them. If the transmission power times the traffic density is large enough, the gateway will consider relocating to a new position. Such consideration does not mean that the gateway will certainly move. The gain for the potential relocation is further qualified against the overhead imposed on sensor nodes by the gateway's relocation.

Secondly, we consider relocation of the gateway in the presence of real-time traffic. We present a novel relocation mechanism for enhanced on-time data delivery in sensor networks. When the average rate of deadline misses for real-time packets starts to increase, our approach pursues moving the gateway to a better position in order to maintain the same or even better level of timeliness. To determine the new gateway's position, we identify locations on or close to the most heavily loaded last-hop node and try to split the incoming traffic passing through that node without extending the delay experienced by real-time packets over other routes. As long as the gateway remains within the transmission range of all the last-hop nodes, we maintain the same routes that were set initially. If it is expected that the new location will put the gateway out of the transmission range of some of the last hop nodes in the current routes, new forwarder nodes that are not involved in any routing activity are selected. Such unused nodes will introduce very little queuing delay, which is desirable for on-time delivery of all real-time packets that use these nodes as relays. If suitable forwarder nodes cannot be found, relocation is either not considered or a new network topology is set at the new location depending on the overhead that rerouting will introduce relative to the gain in timeliness.

In the balance of this section, we describe our system model and survey related work. Section 2 presents our approach for triggering and handling the gateway repositioning under unconstrained traffic. Section 3 addresses the when, where and how issues of gateway relocation in the presence of real-time traffic. Section 4 is dedicated for the experimental validation of the presented approaches, describing the simulation environment and analyzing performance results. Finally, the paper is concluded with a summary in section 5.

1.1. System Model

A set of sensors is spread throughout an area of interest to detect and possibly track events/targets in this area. The sensors are battery-operated and are empowered with limited data processing engines. The mission of these sensors is dynamically changing to serve the need of a command center. A gateway node, which is significantly less energy-constrained than the sensors, is deployed in the physical proximity of sensors. The gateway is assumed to know the geographical location of deployed sensors. Sensors are discovered through repeated beacons or through surveying the area of interest [15][16]. The gateway is responsible for organizing the activities at sensor nodes to achieve a mission, fusing data collected by sensor nodes, coordinating communication among sensor nodes and interacting with the command node. The gateway node sends to the command node reports generated through fusion of sensor readings, e.g. tracks of detected targets. The command node presents these reports to the user and performs system-level fusion of the collected reports for overall situation awareness. The system architecture for the sensor network is depicted in Fig. 1.

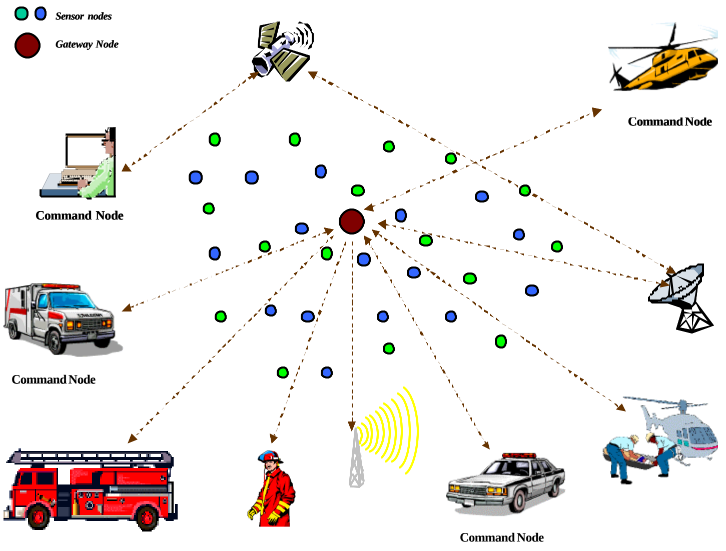


Fig. 1: Three-tier sensor network architecture

While sensor nodes are stationary, we are considering a gateway that has limited motion capabilities. For example the gateway can be a portable computer in the backpack of a rescue crew or on an emergency vehicle. The gateway remains stationary unless the network operation becomes inefficient. The gateway can relocate to another position in order to optimize some performance metrics, e.g. maximizing network lifetime. Sensors are assumed to be within the communication range of the gateway node. The sensor is assumed to be capable of operating in an active mode or a low-power stand-by mode. The sensing, processing and radio circuits can be powered on and off. In addition, the radio's transmission power is assumed to be programmable for a required range. It is worth noting that most of these capabilities are available on some of the advanced sensors, e.g. the Acoustic Ballistic Module from SenTech Inc. [17]. It is also assumed that the sensor can act as a relay to forward data from another sensor. We refer to the gateway's selection of a subset of the sensors for probing the environment as network organization and to the data routing and medium access arbitration as network management.

1.2. Related Work

The limited energy supply of sensor nodes necessitates energy-awareness at most layers of networking protocol stack including the network layer. In addition, many applications of sensor networks require the deployment of a large number of sensor nodes making it impractical to build a global addressing scheme. Moreover, in contrary to contemporary communication networks almost all applications of sensor networks require the flow of sensed data from multiple sources to a particular sink. These unique characteristics of sensor networks have made efficient routing of sensor data one of the technical challenges in wireless sensor networks [12]. While a number of routing protocols pursued a data centric methodology by naming the data [8][9], some considered clustering the sensor nodes in order to decrease the number of transmitted messages to the sink node and have a more scalable setup [7][18]. Other protocols either adopted a location-based routes setup [19][20] or strived to achieve energy saving through activation of a limited subset of nodes [21][22]. In addition, with the increasing interest in the applications that require certain end-to-end performance guarantees, a few routing protocols have been proposed for providing energy-efficient relaying of delay-constrained data [13][14]. To the best of our knowledge the issue of gateway relocation in sensor networks has not been addressed in the literature. While the goals of most published techniques are increasing network lifetime and on-time delivery of data through clever architecture and management of the network, none of the work considered the possibility of relocating the sink (gateway) node for enhanced network performance.

Gateway positioning has been also investigated in the context of wireless local area network and cellular infrastructure [23][24]. The gateways, also called base stations, in these systems are stationary in nature and are placed in order to achieve coverage of an area or a building using the minimal number of gateway units. Another related work to gateway optimal positioning is reported in [25][26]. The considered model is to use the gateway node as a direct router for a group of mobile nodes that would be otherwise unreachable due to topological reasons such as blockages. Unlike our model the gateway's capability to move is not constrained. The problem addressed is to find the optimal place for the gateway node to best serve the group in terms of latency and throughput. The pursed approach is to move the gateway to the weighted geographic centroid of the group by considering the location and traffic generated by nodes regardless the established routes. Determining the gateway position is formulated as an optimization problem. The gateway applies the repositioning algorithm at each sampling instant to gradually get closer to the nodes that generate the highest traffic and to respond to changes in the node locations. Our work is path-based and does not just consider nodes. We argue that it is not possible to optimally place the gateway without considering the network topology and inter-node links. We also observe that frequent changes to the network topology can impose overhead that can surpass the value of the relocation from a system's point of view.

2. Gateway Relocation for Increased Efficiency

Although the energy-aware multi-hop routing, e.g. [18], does dynamically adapt to changes in sensor's energy and traffic pattern, sensors nearby the gateway die quickly as they are the most utilized nodes in the network. As a result, communication with the gateway would use farther hops and the node's energy utilized for communication with the gateway would be considerably higher. Such effect can spread in a spiral manner, thus draining the sensors energy and hence shortening the lifetime of the network. To stop such pattern of energy depletion, we propose relocating the gateway. We consider a model of limited or constrained gateway motion. Example scenarios include when the gateway is a portable computer that can be carried and dropped in another location.

As discussed earlier, the problem of relocating the gateway is a very complex optimization problem due to the infinite solution space and the excessively large number of parameters involved. In addition, a new set of energy-efficient routes has to be established for every interim search point in order to qualify the impact on the optimization objective. Since such complexity can be undesirably a resource burden for the gateway, we follow a heuristic approach instead. The main idea of the search heuristics is to move the gateway towards the sources of largest traffic. Such approach has multiple advantages. First, the gateway will be in the area where more nodes are collecting data and thus communication-related energy consumption will be reduced. It is thus expected that the average energy per packet gets reduced. In addition, nodes collecting the most data will be closer to the gateway and most probably fewer hops will be involved lowering the overall latency time for data collection. Moreover, the packet throughput will be higher since it is expected that messages pass through fewer hops and travel shorter distances making them less likely to be dropped. In summary, such approach for relocating the gateway not only can increase network lifetime but also enhance other performance metrics like latency and throughput.

To achieve such performance gains, one needs to address the basic questions of when to relocate the gateway, where to place it and how to handle data traffic during the transition. In this section we address these three issues for non-constrained traffic basing the analysis on energy metrics. In the next section, we investigate latency-motivated gateway repositioning with a goal of enhancing timeliness of delay-constrained traffic.

2.1. When to Move the Gateway

The decision for a gateway movement has to be based on an unacceptable performance despite the use of the most effective methodology for route setup. Once the gateway detects such unacceptable performance it will consider moving to a new position. Such consideration does not necessarily lead to an actual relocation. The gateway qualifies the impact of the potential new position on the quality of the route setup, measured in some static metric. If the impact is positive, the gateway starts to move. Therefore the “When” and “Where” issues of the gateway movement are very closely inter-related. The following notation is used in deriving the condition for beneficial gateway relocation:

G : Gateway node

S_1 : Set of sensors which are less than distance D away from the gateway

S_R : Set of sensors selected by the routing algorithm which are one hop away from the gateway on the current active set of routes

R : Reachable range of the transmission of the nodes in S_R

S_{R1} : Set of sensors in S_R which are also in S_1, i.e. $S_{R1} = S_1 \cap S_R$

S_1^{new} : Set of sensors which are less than distance D away from the gateway when placed at a different position than the current one

S_R^{new} : Set of sensors selected by the routing algorithm which are one hop away from the gateway on the new route when placed at a different position than the current one

S_{R2} : Set of sensors in S_R^{new} which are also in S_1^{new}, i.e. $S_{R2} = S_1^{new} \cap S_R^{new}$

P_i : The packet traffic measured as the number of packets per frame, going through a node i

PT : A set consisting of packet traffic of each sensor in S_{R1} in an ascending order

PT^{new} : A set consisting of packet traffic of each sensor in S_{R2} in an ascending order

$E_{(TRi)}$: Energy consumed by a node i in transmission of a packet to the next hop

Note that S_1^{new}, S_R^{new}, S_{R2} and PT^{new} are calculated by placing the gateway at the new location. The basic idea is for the gateway to check the changes in data paths in consecutive routing cycles. Typically re-routing is performed in response to a high packet loss caused by the energy depletion or failure of a relay node, or is triggered by a change in data sources that requires setting a new topology [18]. When comparing routes in two consecutive cycles, if the data sources are the same and the nodes in previous S_{R1} differs from that the current S_{R1}, the gateway perform further analysis. The gateway checks the nodes that used to be in the previous S_{R1} and are excluded from the new S_{R1}. If these nodes were among the bottom x% of the set PT, relocation would not be necessary. On the other hand if these nodes were forwarding high traffic (among top (1- x)% in PT), the gateway perform a heuristic search for a better position. It should be noted that the x% cutoff depends on the data generation rate.

To qualify the impact of placing the gateway at a new position, some static network performance metric have to be used. Most of the popular performance metrics for sensor networks, such as average delay per packet and throughput, are based on the network operation over a period of time and are typically network-wide in scope. Therefore, we use as a metric the total transmission energy of the relay node that are one-hop away from the gateway, basically those in S_R and S_R^{new}. A positive impact on the total transmission energy for these nodes would be a good indication for the gateway motion. However, the reduction in the total transmission energy has to exceed an application threshold to justify the overhead, as discussed later in this section. A constant D can be derived based on the overhead for handling the motion both at the gateway and at the network level. The condition for relocation can be mathematically defined as:

$$\sum_{\forall i \in S_R} E_{(TR_i)} \times P_i - \sum_{\forall j \in S_R^{new}} E_{(TR_j)} \times P_j > \Delta$$

As mentioned earlier, finding an optimal new location for the gateway is an implicit issue for the relocation decision. Optimal placement of the gateway is an NP-hard problem in the general term. Therefore we purse a heuristic search and settle for a sub-optimal location to overcome such complexity. In the next subsection we explain such heuristic approach.

2.2. New Gateway's Position

Determining the new position to which the gateway would relocate is the central problem. To limit the complexity of such optimization we determine few directions for the gateway motion based on the current network traffic. The idea is to move the gateway towards the sensors that generate the most number of packets. However, it may be infeasible for the gateway to go very far from its current position due to limited mobility resources or harsh terrain. Moreover, a big move in one direction may cause the gateway to lose connectivity to other data sources; an issue that is not easy to sufficiently handle by extending the path from these sources due to a shortage in the available live nodes in S_1 that can act as forwarders. In addition, the topology of the sensor network changes dynamically and thus there is a risk for wasting substantial resources for reaching a far position, which turns shortly after to be not optimal due to changes in the environment. To achieve similar effect, we try placing the gateway close to the relay nodes in S_R, which routes the largest number of packets. To identify such relay nodes, the list PT is consulted and the top nodes on the list are picked as candidates.

If the top node in PT is dominating in terms of traffic density, the gateway is to move towards that node. For simplicity we use a linear motion pattern, i.e. the gateway move in a straight line. In case multiple relay sensors share that dominance in packet traffic, a weighted average based on the distance between the gateway and these relay sensors and their traffic density is pursued. The idea is to balance the gateway's interest while picking the direction of the move. A position that is equidistant, in terms of distance × traffic density, from the relay sensors on the different high traffic paths is calculated. Finding such a position is similar to identifying the centroid of a polygon consisting of multiple points. An algorithm for identifying the centroid is sketched in Appendix B [38]. We have studied the implication of the number of selected relays nodes (polygon corners) on

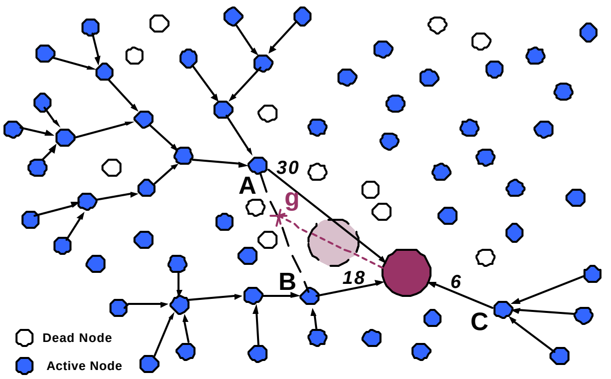


Fig. 2: The gateway balances its interest in serving nodes A and B according to their proximity and traffic density while setting the direction of motion and also considers node C's transmission range when deciding on how far it should travel in that direction.

performance, in terms of energy and throughput, through simulation. The results, as we discuss in section 4, indicates that considering the nodes up to the median of *PT* tends to results in the best performance gains.

While positioning the gateway at the centroid will be ideal for high traffic paths, it can worsen the total power consumed on other paths with lower traffic density. Therefore, the gateway will validate the overall impact on transmission energy before confirming the move. The expected network topology at the new location is checked to ensure that the total transmission energy of nodes in S_R^{new} is significantly less than those of S_R. Fig. 2 illustrates how to pick the direction of the gateway movement. The traffic density is annotated next to the link between the nodes A, B, C and the gateway. Nodes A and B are the top two relay nodes in *PT*. Balancing the interest of the relay nodes *A* and *B* changes the slope of the gateway travel path. However, moving the gateway at the position *g* may get it out the transmission range of node *C*. Therefore the gateway has to ensure that node C stays reachable by increasing its transmission power or by involving forwarder nodes, as we explain next.

2.3. Handling Packet Delivery during the Move

Once the new location of the gateway has been picked and confirmed to enhance the performance of the sensor network, a plan for moving the gateway without losing the packets destined to it has to be generated. Our gateway motion handling mechanism tries to maintain continual packet delivery to the gateway while restricting changes to the network topology. Avoiding radical changes to the data routes limits the disruption of on-going data traffic and also curtails the overhead that the gateway's relocation introduces to the network. We pursue two alternative approaches based on the current network state. In the first approach the transmission power of some sensors is adjusted to ensure that data packets can reach the gateway (Fig 3). The second approach designates some forwarders to extend the current routes (Fig. 4).

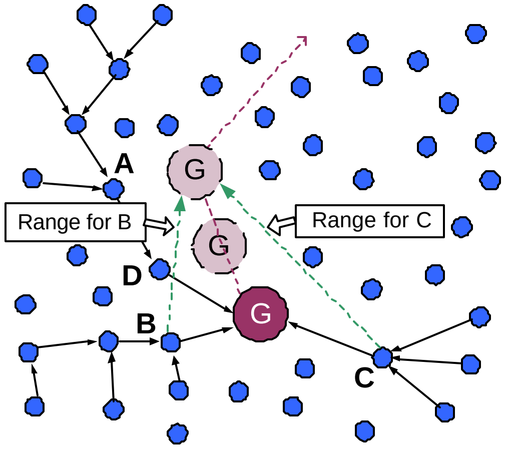


Fig. 3: Range of nodes that directly communicate with the gateway is increased to ensure uninterrupted message delivery until the gateway reaches the next intermediate position

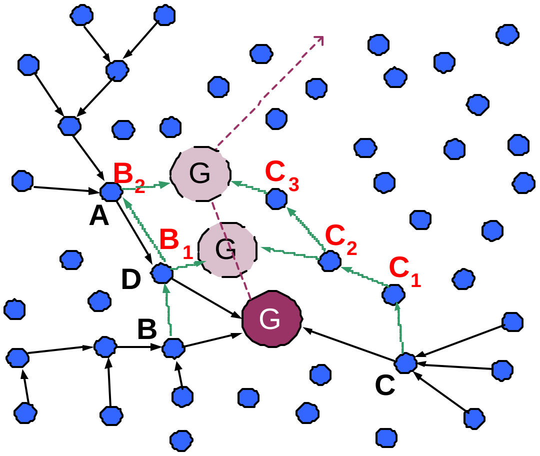


Fig. 4: Sensors C_1, C_2 and C_3 act as relays to forward messages from C during gateway motion. Sensors D and A act as relays B_1 and B_2 respectively for messages from B.

The gateway is assumed to move towards its new position in strides. The strides are to be in straight line between intermediate points. At each intermediate point, the gateway checks whether it can still be reachable by the last hops while traveling on the next stride. If the gateway at the new intermediate point stays within the transmission range of the last hop nodes, the gateway will simply indicate to these nodes the need for adjusting the output power of their radio to cover the gateway's next move. The gateway can then start approaching the new intermediate point without worrying about the potential of broken links. If the gateway detects that it can lose connectivity to the last hop nodes while on the move, it explores the option of employing sensor nodes to forward the packets. Once, it assesses the need for employing forwarder nodes, the routes are extended by those forwarders and that information is sent to relevant sensor nodes. This procedure is repeated till reaching the new position set by the relocation algorithm.

2.4. Pseudo Code and Complexity Analysis of the Algorithm

The proposed algorithm for the relocation is depicted in Fig. 5. Once an unacceptable performance is observed and the gateway relocation is considered in line 1, we identify *m* nodes that are one hop away from the gateway and relay the highest data traffic in line 2. The new location for the gateway is estimated to be equidistant, in terms of distance × traffic density, from these *m* nodes (line 3). As mentioned earlier, this is similar to the problem of finding the centroid of *m* points. It should be noted that if the relocation is geared for continual performance enhancement, line 1 will be simply a time-based activation. The value of *m* can be set to be as simple as 3 and can be as high as the $|PT|$. We study the choice of *m* experimentally in section 4. In the next step, line 4, the routing algorithm is run based on the assumption that the gateway is placed at the new location found in line 3. In order to qualify the impact of relocation, total sensors' transmission power for the old and new gateway locations is calculated in lines 5 and 6 respectively. In line 7, we check if the gain in terms of transmission power is more than a certain threshold. If this is the case, the gateway is moved to the new location (line 8), otherwise it remains at the old location (line 9).

```
RelocationAssess()
1  if an unacceptable performance is detected consider relocating the Gateway (G)
     /* Find a new location to move G */
2    Identify the m nodes $k_1, k_2, k_3, .., k_m$ with the highest Packet traffic going through it.
3    Find Pos($G_{new}$)=Centroid($k_1, k_2, k_3, .., k_m$)  /* Using centroid computation */
4    Get the new routes and the set $S_R^{new}$ based on Pos($G_{new}$)
5    $TPower_{Old} \leftarrow \sum_{\forall j \in S_R} E_{(Tr_j)} * P_j$  /* Compute total Power at old position */
6    $TPower_{New} \leftarrow \sum_{\forall i \in S_R^{new}} E_{(Tr_i)} * P_i$  /*Compute total power at new position*/
7    if ($TPower_{Old} < TPower_{New} + \Delta$) /* $\Delta$ is the threshold */
8        Relocate (G, Pos($G_{new}$))
9  else Remain at the same location

Relocate(G, Pos)
10 While (Pos is not reached)
11     Proceed one stride
12     for ($\forall j \in$ LastHopNodes) /* The list of nodes within one hop neighbor of G */
           /*TRange is the transmission range of a sensor, $G_{next}$ is the next location of G */
13         if (Dist(j, $G_{next}$)>TRange[j]) /* The gateway will get out of range */
14             FindForwarder(j); /* Find another node to relay the packets */
15         else TransmissionRange[j]$\leftarrow$ Dist(j, $G_{next}$);

FindForwarder(j, $G_{next}$)
     /*Find a relay node i between the next position of G and node j */
16  Find Forwarder i¦ {Dist(j,i)<TRange[j] && Dist(i,$G_{next}$)<TRange[i] && Remain_energy(i) > d }
17  Update RouteTable[j];
18  if (i not in LastHopNodes) /*Update the list of LastHopNodes accordingly*/
19    begin
20      Add i to LastHopNodes
21      UpdateRouteTable[i];
22    end
23  Remove j from LastHopNodes
```

Fig. 5: Gateway relocation heuristics for enhancing the energy-efficiency of network operation

When the gateway starts to move towards its new position (lines 10-11), at each stride, the algorithm checks each sensor in the list of nodes that are one hop away from the gateway (i.e. LastHopNodes) to identify those which the gateway will get out of their range at the next intermediate position namely G_{next} (lines 12-13). For each of those nodes, a forwarder is found in line 14. If the gateway stays reachable, the last hop node adjusts its transmission power so that the gateway receives the messages uninterruptedly until reaching the next intermediate position (line 15). Line 16 reflects the criteria for selecting the forwarder; basically the forwarder has to reach both the gateway and the considered last hop node "*j*" while having sufficient energy to play its role. In case of multiple alternatives, the one with the larger remaining energy is picked. The routing table for the considered last hop node "*j*" is updated in line 17. If the forwarder node "*i*" is not a member of the list *LastHopNodes*, it will be added in lines 18-20 and its routing table is updated accordingly in line 21.

In order to assess the complexity of the relocation algorithm, we consider the number of iterations in the function "*Relocate*" (line 10) and the function "*FindForwarder*" (line 16). *Relocate* checks the last hop nodes that directly communicate with the gateway in order to identify any node that may not be able to reach the gateway during the next move. Assuming the number of nodes in *LastHopNodes* is *l*, the number of iterations to skim through that list will be $O(l)$. When the gateway is moved to a new location and loses connection with some of the nodes in *LastHopNodes*, a forwarder is to be found for each of such nodes. In order to find such forwarders, $l*(n-1)$ iteration is made since in the worst case a forwarder needs to be identified for every node in the *LastHopNodes* list and all deployed sensors other than the node under consideration may have to be checked for being a potential forwarder. Therefore, the total number of iterations will be $l + l*(n-1)=l*n$ which will be $O(n^2)$ in the worst case. We argue that such complexity would not make relocation of the gateway a major resource burden.

3. Enhancing Timeliness of Delay-Constrained Traffic

In the previous section, we have addressed the issues of gateway relocation with an objective of minimizing the energy consumption and extending the network lifetime. While our approach is expected, and also confirmed by our experiments discussed in section 5, to decrease the average delay per packet, it does not always improve timeliness of traffic that is constrained in end-to-end delay. Favoring motion in the direction of high sources of data may extend the delay experienced by time-constrained data, which happen to be generated by sensors, topologically at the opposite end. In this section, we study the issues of gateway relocation that is motivated by diminishing timeliness of delay-constrained traffic.

Recently, there has been a growing interest in sensor networks applications that require certain performance guarantees such as end-to-end delay. For example, routing of imaging data in a battle environment requires careful handling in order to ensure that the end-to-end delay is within acceptable range and the images are received properly without any distortion. Other typical applications include real-time target tracking, emergent event triggering in monitoring applications and critical information relaying in emergency applications. Routing sensor data in these applications not only should pursue the most energy efficient path but should also ensure soft real-time delay guarantees over the duration of a connection [13][14].

The location of the gateway becomes influential when real-time traffic is involved. For instance, when routes to the gateway get congested, most requests for establishing paths for real-time data may be denied or the deadline miss rate of real-time packets may increase significantly. Traffic congestion can be caused by an increase in the number of real-time data packets coming from nodes close to a recent event. In such circumstances, it may not be feasible to meet the requirements for real-time data delivery. If the gateway has a capability to move, it can be beneficial to relocate the gateway close to the occurring event in order to spread the traffic on additional hops and to increase the feasibility for meeting the real-time delivery requirements. It should be noted that this scenario cannot be handled using our approach described earlier since there may be unconstrained traffic that leads the motion to a different direction, or even deems the relocation useless.

Based on the observations above, we define the problem as follows: Given the location of the gateway and currently established routes for delay-constrained data, we are interested in relocating the gateway to a new position such that lower deadline miss rate is achieved without a significant increase in the energy usage in the network. As stated in section 2, exhaustive search algorithms are too complex and resource demanding and thus are not suitable for sensor networks. Instead a heuristic approach is pursued. In this section, we first describe the underlying routing mechanism for real-time data and then explain our approach for when, where and how to relocate the gateway for increased timeliness.

3.1. Real-time Routing Mechanism

The routing approach for supporting real-time traffic is based on the model defined in [14]. Each node employs a packet scheduling discipline that approximates Generalized Processor Sharing (GPS) [27]. GPS achieves exact weighted fairness by dedicating a separate FIFO queue for each session (flow) and serving an infinitely small amount of data from each queue in a weighted round robin fashion. The packetized version of GPS is called Weighted Fair Queuing (WFQ). One interesting property of WFQ is that when combined with leaky bucket constrained sources, it can provide upper end-to-end delay bounds for each flow [28][29]. Assuming flow *i* is constrained by a leaky bucket with parameters (s_i, r_i), the maximum end-to-end delay (transmission + queuing delay) for a packet of flow *i* under WFQ, given in [29], is:

$$D(i) \le \frac{s_i}{g(i)} + \sum_{m=1}^{M-1} \frac{P_{\max}(i)}{g_i^m} + \sum_{m=1}^{M} \frac{P_{\max}}{C} \qquad [1]$$

$$g_i^m = \frac{\Phi_i^m}{\sum_{j=1}^{n} \Phi_j^m} C \qquad [2]$$

where,

- *C* is the outgoing link bandwidth
- *$P_{max}(i)$ is* maximum packet size for flow i
- *P_{max} is* maximum packet size in the network
- *g_i^m* is the service rate on node m for flow i
- *g(i)* is the minimum of all service rates for flow i
- *M* is the number of nodes on path of flow i
- ? $_i^m$ is the link share on node m for flow i

The approach of [14] uses an approximation of WFQ by considering each imaging sensor node as a source of different real-time flow with only one real-time queue to accommodate the real-time data coming from these multiple flows (Fig. 6). This model is used due to two reasons: First, having a different queue for each real-time flow will be inefficient given the storage capacity of a sensor node. Second, the real-time flows are generated dynamically depending on the number of active imaging sensors. Since the number of such flows can change during the sensing activity, having one queue will reduce the maintenance overhead.

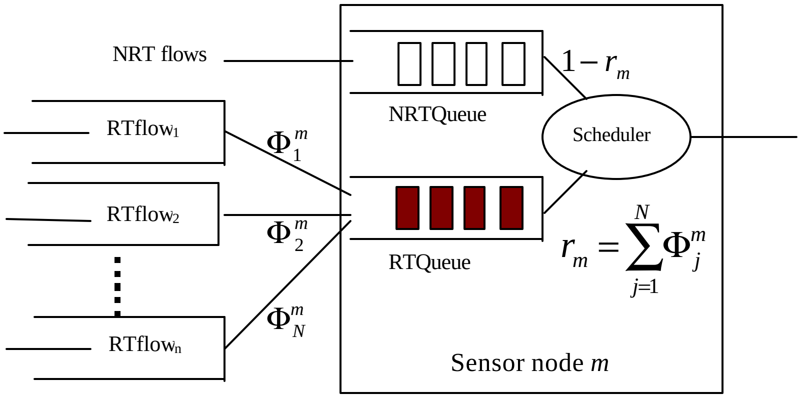


Fig. 6: Queuing model on a sensor node

The service ratio “*r*” is the bandwidth ratio set by the gateway and is used in allocating the amount of bandwidth to be dedicated to the real-time and non-real-time traffic on a particular outgoing link. This value is also used to calculate the service rate for each type of traffic on that particular node, with $r_m m$ and $(1 - r_m)m$ being respectively the service rate for real-time and non-real-time data on sensor node *m*. In this case, r_m for the real-time queue on a node is the summation of link shares ($\Phi_1^m, \Phi_2^m, ..., \Phi_n^m$) of all real-time flows passing through that node as shown in Fig. 6. Each link share is calculated through the formula (2).

In order to setup real-time routes to the gateway, the approach first obtains a set of energy-efficient paths for each real-time source sorted in an ascending order with respect to the energy consumed per packet. These paths are further checked, starting from the most energy efficient one, in order to identify the path that can meet the end-to-end delay requirement. A path meets the timeliness requirement if an appropriate *r*-value can be set for each node on that path. Finally, the real-time connection is established for the picked path.

3.2. Gateway Relocation Approach

Again the relocation approach has to address the three issues: when to relocate the gateway, where to place it and how to handle its motion without disrupting the network traffic. The following analyzes these issues and describes our algorithm.

Assessing gateway's relocation: Given the delay-constrained nature of the real-time traffic, the need for relocation can be observed by monitoring a timeliness-based metric such as the deadline miss rate. In the underlying routing mechanism the miss rate is indirectly related to finding proper *r*-values. Recall that, the mechanism for real-time routing basically tries to find feasible *r*-values by considering every path from sources of real-time data to the gateway. In cases where proper *r*-values between 0 and 1 cannot be found, the connection is simply rejected and the path is not established. Moreover, even when proper *r*-values are found, the miss rate can start to grow due to detection of new events that cause the generation of real-time packets at a higher rate than planned and thus leading to increased congestion. Therefore, the trigger for relocation can be the unacceptable increase in the miss rate of real-time packets. In our approach, the gateway will set a threshold for the maximum level of miss rate by maintaining such statistics periodically and consider relocating the gateway to a better position when such threshold is reached. Since some deadline misses can result from packet retransmitting caused by transient noise, instantaneous relocation of the gateway would not help much. Therefore, setting a threshold will ensure that the misses are persistent and resultant from high traffic on some relaying nodes. It is worth noting that the gateway can be relocated more than once whenever necessary during the data traffic cycles since our handling of the gateway motion, discussed later in this section, strives to preserve the current route setup. We also like to note that the assessment of the gateway's relocation can be conducted periodically; motivated by a continual quest for improving the network performance as explained in section 2.

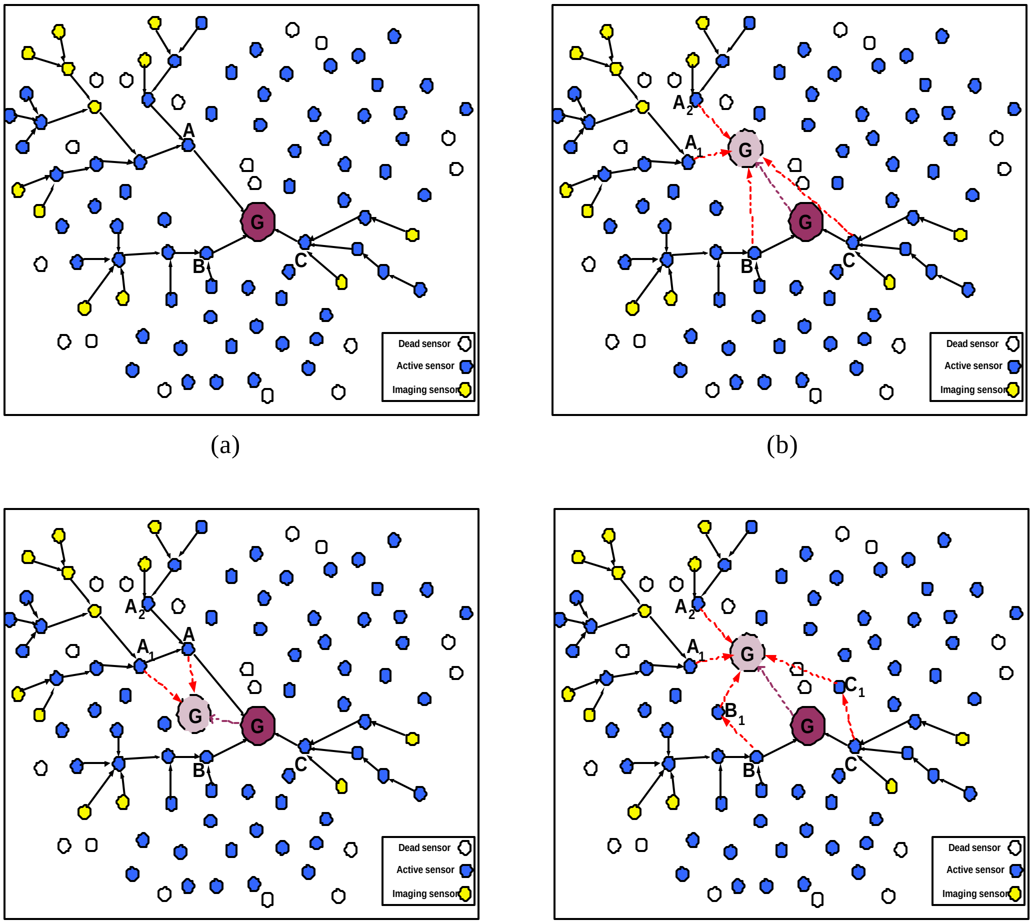
Finding new gateway's position: After deciding that the gateway is to be relocated, the next step is to identify a better position for the gateway. The main aim here is to move the gateway towards the loaded nodes in terms of real-time traffic so that the end-to-end delay can be decreased. Therefore, our approach first browses the set of last hop nodes, i.e. the nodes that directly transmit to the gateway, in order to pick the hop with the biggest *r*-value and consider relocating the gateway at the position of that hop. This strategy brings two advantages: (1) it can decrease the average end-to-end delay since real-time data packets have to travel a fewer number of hops; (2) it will allow admitting more real-time flows since the load is alleviated by splitting the traffic.

However, identifying a new position for the gateway does not necessarily mean that the gateway will be relocated. Before moving the gateway, an analysis is performed in order to assess the potential overhead that will be introduced. Such overhead will be due to any necessary route adjustment when the gateway goes out of the transmission range of some of the other last hop nodes. If this is the case, additional relay nodes should be identified to maintain uninterrupted data delivery and consequently some routes will need to be updated accordingly. While involving such additional relay nodes will boost the end-to-end delay for real-time packets and increase energy consumption, forming a new network topology based on the new position of the gateway can be undesirable due to the control traffic overhead and the disruption of data collection [18][30]. In addition to handling the loss of connectivity, we do not consider larger movements for the gateway in order to avoid excessive relocation is such dynamic environment, as also stated in Section 2.2. Instead of making big moves, we pursue a cautious approach in order to ensure that the network topology stayed adjustable without disrupting data delivery. It is worth noting that the gateway can be further relocated as deemed necessary.

Therefore, when a new position is nominated, our approach first checks whether at the new location the gateway will go out of the transmission range of the last hop nodes on the current routes. This is also used for breaking the ties when multiple alternative nodes with the same *r*-value are found. In such circumstances, if the other last hop nodes can still reach the gateway by increasing their transmission power, relocation can be performed safely. A pictorial illustration of this situation is depicted in Fig. 7a and 7b. If we assume that there is an occurring event on north-west of node A, many of the imaging sensors that are sources of real-time traffic will be turned on in that region, increasing the traffic density flowing through node A (Fig. 7a). In this case, most of the routes passing through node A will be rejected since the limited service rate on node A increases the end-to-end queuing delay. However, the load on node A can be alleviated by moving the gateway to the location of A since the queuing delay at node A will be completely avoided. If the gateway is still reachable by nodes B and C, they just increase their transmission power for uninterrupted data delivery as seen in Fig 7b.

If the new location requires some topology changes, our approach considers alternative positions that might be feasible for the gateway to be positioned at without causing any effect on the network topology. Such alternatives include the location of one of the siblings of the node with the biggest *r*-value or some other location that will be closer to that sibling and enable data relaying without affecting the latency of other real-time packets. Here the selection of the sibling node is decided based on its *r*-value since targeting the sibling with the biggest *r*-value can help best in splitting the real-time traffic and hence decrease the queuing delay. An example of this situation is shown in Fig. 7c. In this case, A_1, which is the sibling of A with the biggest *r*-value, is picked and the gateway is relocated closer to that node so that the load of A is split between A and A_1. Such relocation does not require any route updates for the current topology. Note that, the new position of the gateway should be within the communication range of A_1, A, B and C so that the traffic passing through A_1 can be directed to the gateway and at the same time nodes A, B and C can still reach the gateway without any need to adjust the routes.

In cases where a new location that will keep the gateway in range cannot be found, our approach strives to minimize the overhead of adjusting the routes. New forwarder nodes that are not currently part of any routes



(a) (b)

(c) (d)

Fig. 7: (a) Initial routes (b) Gateway is relocated to the location of A if it is not out of range of C and B. (c) If that is not possible, the gateway is relocated to a location close to B. (d) Otherwise unused forwarders B_1 and C_1 are found for relaying of traffic from B and C

of real-time data are picked for each of the last hop node in order to provide uninterrupted delivery of real-time packets. Since such nodes will not introduce extra queuing delay for real-time data, this will not affect the end-to-end delay of real-time data using that path. An example is shown in Fig. 7d. In this case, nodes B_1 and C_1, which are not part of any existing routes, are designated as forwarders for nodes B and C respectively. Failure to find such forwarders will result in keeping the gateway in its current position since the possible gain in relocation can be degraded due to increased delay on some routes, as explained above. It should be noted that in some network architectures the gain achieved by the gateway relocation is valued more than the rerouting overhead. In such architectures it would make sense not to give up on gateway relocation if the network topology cannot be maintained. Since this scenario depends highly on the routing protocol and network management strategy, we have decided to flag such case in our algorithm for further architecture specific analysis or to simply seize the gateway movement.

Managing the Move: The continual delivery of data packets while the gateway is on the move is provided through the same mechanism discussed in the previous section when no real-time traffic is involved. However, when we explore the option of employing forwarder nodes, the addition of new relays can increase the end-to-end delay and cause some packets to miss their deadlines. Therefore, ideal forwarder nodes should not be currently involved in relaying real-time traffic as stated earlier.

3.3. Detailed Algorithm

The proposed algorithm for gateway relocation is shown in Fig. 8. The gateway monitors the miss rate for real-time data periodically to detect situations when there is a need for relocation (line 1-2). Again, the motive for relocation can also be a desire to improve timeless of packet delivery and not just a reaction to unacceptable deadline misses. If the gateway motion is justified, a new position is identified (line 3) and the gateway starts to move to its new location (lines 4-5). In order to designate a new location for the gateway, the algorithm finds the location of the node *m* with the biggest *r*-value in line 6. In case of ties, the closest node to the gateway will be picked. Once the new location is identified, the algorithm checks in line 7 whether moving to that location will require employing forwarder nodes or not. The need for forwarders can be determined by comparing the distance to the new location and transmission ranges of the nodes in *LastHopNodes*. Since the inclusion of forwarders can increase delay on the route, alternative locations are searched in line 8. In this case, we consider a location that will be within the transmission range of the sibling node of *m* that has the largest *r-value* and at the same time will keep connectivity with other nodes in the *LastHopNodes* (line 9). If no such location is available, it means that employing forwarder nodes is unavoidable and the algorithm assumes the new gateway position as the location of node *m* and looks for forwarder nodes that are not currently part of any data paths (lines 10 and 14). When such forwarders cannot be identified (line 12), relocation will not be performed and new position of the gateway remains the same as stated in line 13. It should be noted that an analysis specific to the network architecture and the detailed routing protocol can be conducted at this stage to further check the suitability of the move by comparing the significance of the gains of repositioning the gateway to the overhead incurred in setting new topology at the new location. Such analysis can be easily augmented. Finally the *SearchForNewLocation* function returns the new location of the gateway in line 16.

When finding the forwarder nodes, the algorithm uses the criteria given in line 17. That is, a forwarder node with no real-time data load (i.e. *r-value* is 0) and having enough remaining energy is favored. The routing table for the considered last hop node *j* is updated in line 18. If the relay node *i* is not a member of the list *LastHopNodes*, it is basically added to that list in lines 19-20 and its routing table is updated accordingly in line 21. Finally, *j* is removed from the *LastHopNodes* in line 22 since it is not an element of that list anymore.

Similar to section 2.4, a complexity analysis can be conducted for the algorithm. We consider the number of iterations in functions "*SearchForNewLocation*" in line 3 and "*FindForwarder*" in line 11 of the algorithm. *SearchForNewLocation* checks only *r-values* of the last hop nodes. The remaining parts of the algorithm will incur the same number of iterations and thus will be $O(n^2)$ in the worst case.

```
Assess_Timeliness()
Check the miss rate at a predefined period
if (Miss rate > Threshold)
        SearchForNewLocation()
if (NewLocation(G) ¹ Pos(G))
        Start moving gateway to NewLocation(G)

point SearchForNewLocation()
Find the node ¦ $m \leftarrow \max_{(j \in LastHopNodes)} (r-value^{j})$ /*LastHopNodes: within one hop neighbor of G*/
  /*Check if gateway goes out of transmission range for any of the nodes in LastHopNodes when
  relocated at the position of node m*/
if $(\exists j \mid TRange[j] < Dist(Pos(j), Pos(m))\ \&\&\ j \in LastHopNodes)$
  begin
        /*Consider moving it closer to one of m's siblings having the biggest r-value*/
        Find the intersecting point IP s.t.
        $\{TRange[sibling] > Dist(sibling, IP)\ \&\&\ TRange[j] > Dist(j, IP) \mid \forall j \in LastHopNodes\}$
        NewLocation(G)= IP;
        if (no IP is found) /* G still goes out of transmission range of some nodes in LastHopNodes*/
          begin
                /* Relocate to Pos(m) by employing forwarder nodes*/
                FindForwarder(j);
                if (no forwarder can be found) /* Do not relocate */
                    NewLocation(G)←Pos(G);
                NewLocation(G)←Pos( m);
          end
  end
else NewLocation(G)← Pos(m);
return NewLocation(G);

FindForwarder(j)
    /*Find a relay node i between the new position of G and node j which is free */
Find Forwarder i ¦ {Dist(j,i)<TRange[j] && Dist(i,$G_{new}$)<TRange[i] && r-value$^{i}$==0 &&
Remain_energy(i) > d }
Update RouteTable[j];
if (i not in LastHopNodes) /*Update the list of LastHopNodes accordingly*/
  begin
      Add i to LastHopNodes
      UpdateRouteTable[i];
  end
Remove j from LastHopNodes
```

Fig. 8: Relocation algorithm for the gateway under real-time traffic

4. Experimental Validation

The effectiveness of repositioning the gateway is validated through simulation. This section describes the network operation, simulation environment, performance metrics, and experimental results.

4.1. Network Operation and Experiment Setup

We have adapted the network operational model of [18] for validating our approach. The network is tasked with a target tracking application with the gateway assuming responsibility for sensor organization based on missions that are assigned to the network. Mission-oriented organization of the sensor network enables the appropriate selection of only a subset of the sensors to be turned on and thus avoids wasting the energy of sensors that do not have to be involved. Thus the gateway controls the configuration of the data processing circuitry of each sensor. The sensor nodes can be in one of four main states: sensing only, relaying only, sensing-relaying, and inactive. In the *sensing state*, the node turns on its sensing circuitry and sends data to the gateway at a constant rate. In the *relaying state*, the node does not probe the environment but its communication circuitry is on to relay data from other active nodes. When a node is both probing the environment and relaying messages from other nodes, it is considered in the *sensing-relaying state*. Otherwise, the node is considered *inactive* and can switch to a low power sleep mode.

A time division multiple access (TDMA) based MAC protocol is employed. The gateway manages slot assignment based on the network topology. The gateway informs each node about the slots in which it should listen to other nodes' transmission and the slots that the node can use for its own transmission. This eliminates the possible interferences during transmission and reception. In addition, the gateway broadcasts the routing table to all sensors prior to starting or resuming data transmission. Routes are set by using the following cost function for a link between nodes *i* and *j*:

$$cost_{ij} = c_0 \times (dist_{ij})^2 + c_1 \times f_1(energy_j) + c_2 \times f_2(e_{ij})$$

where,

- c_0, c_1 and c_2 are constants.
- $dist_{ij}$ is the distance between the nodes *i* and *j*.
- $f_1(energy_j)$ is the function of the current residual energy of node *j*, and is taken as (1/ $energy_j$) favoring nodes with rich energy reserve.
- $f_2(e_{ij})$ is the function of the factors that affect the error rate on the link between *i* and *j* (i.e.. Links with high error rate will increase the cost function, thus will be avoided). f_2 is approximated as $dist_{ij}$ / $buffer\_size_j$.

For a node in the sensing state, packets are generated at a constant rate of 1 packet/sec. This value is consistent with the specifications of the Acoustic Ballistic Module from SenTech Inc. [17]. Each data packet is time-stamped when it is generated to allow the calculation of the average delay per packet. In addition, each packet has an energy field that is updated during the packet transmission in order to calculate the average energy per packet and to track the remaining energy at each node. For real-time traffic generation, we assume that each sensor node is capable of taking the image of the target to identify it clearly and can turn on its imaging capability on demand. During the simulation, small subsets of the active nodes, which are the closest to the target, turn on their imaging capabilities. Therefore, the imaging sensors may change with the movement of the target. The packet generation rate for imaging sensors is bigger than the normal sensors. Packets, generated by imaging sensors, are labeled as of real-time type and treated differently at the relaying nodes. The *r*-value is initially assumed to be 0 but it is recalculated as imaging sensors get activated. The default end-to-end delay requirement for real-time data is taken to be 0.08 sec [31]. The energy model of [36] is used for transmission and reception energy costs. No energy consumption is assumed for the sleep mode. A more elaborate description of the used energy model can be found in Appendix A. The algorithm used for calculating the centroid, required when applying the relocation algorithm in Fig. 5, is summarized in Appendix B.

In the experiments, the network consists of varying number of sensor nodes (50 to 200) that are randomly placed in a 500×500 m^2 area. The gateway initial position is determined randomly within the region boundaries. A free space propagation channel model is assumed [32] with the capacity set to 2Mbps. Each node is assumed to have an initial energy of 5 joules. A node is considered non-functional if its energy gets completely depleted. The maximum transmission (power) range for a sensor node is assumed to be 50 meters [33]. We assume in the experiment that the network is tasked with a target tracking mission and the gateway can move with a maximum speed of 5 m/sec when needed. Targets are assumed to come from outside the area. Therefore, the initial set of sensing nodes is chosen to be the nodes on the convex hull of sensors in the deployment area. The sensing circuitry of the other nodes is usually turned off but can be turned on according to the target location. Targets are characterized by having a constant speed chosen uniformly from the range 4 m/s to 6 m/s and a constant direction chosen uniformly depending on the initial target position in order for the target to cross the convex hull region. Any target remains active until it leaves the deployment region area.

4.2. Performance Results

In this section we present some performance results obtained with and without repositioning the gateway. We use the following metrics to capture the performance of our approach:

- *Average energy per packet:* This metric represents the average energy consumed in transmitting and receiving a data packet.
- *Average lifetime of a node:* This metric gives a good measure of the network lifetime by averaging the time a node stay functional.
- *Average delay per packet:* Defined as the average time a packet takes from a sensor node to the gateway.
- *Deadline miss rate*: This is one of the most important metrics in real-time applications, which indicates the percentage of lost or tardy packets compared to the total number of packets generated.
- *Network Throughput:* Defined as the total number of data packets received at the gateway divided by the simulation time.

In the experiments, we have applied 5 distinct seeds in order to generate random network topologies. Separate simulation experiments were performed for each topology. We observed that with 90% confidence level, the simulation results stay within 6%-10% of the sample mean.

4.2.1 Relocation under Unconstrained Traffic

We have first considered relocation of the gateway under unconstrained traffic. In this case, the baseline is the routing approach that uses the same link cost while employing a stationary gateway. To identify the nodes that are to be considered when determining the new position, the best value of "*m*" in the relocation algorithm, we have conducted numerous simulation experiments with varying values of *m*. The experimental results have indicated that the performance in terms of node's lifetime and throughput, improves significantly as the number of nodes exceeds 3. When "*m*" was again boosted, such improvement continued, though at a slowing pace, until reaching the median of *PT*. Beyond the median, the performance gain starts to diminish due to the consideration of lightly-loaded nodes. Fig. 9 shows the effect on "*m*" on the average node life time for a network of 100 sensors. Therefore, we have considered only the top nodes in *PT* up to the median when calculating the next position. We note that during the simulation the gateway relocated 11 times and traveled on the average a distance of 27.1m per relocation.

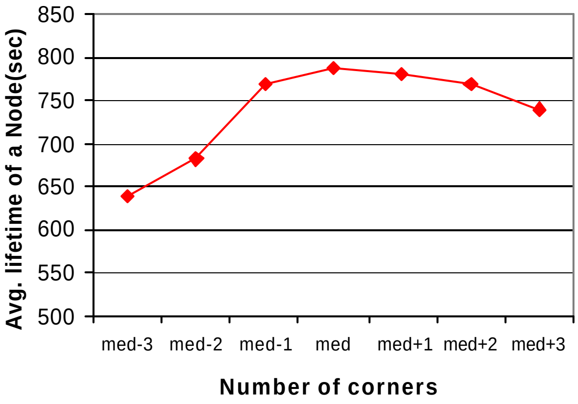


Fig. 9: Effect of varying the number of nodes considered in determining the gateway new location on the gain in network lifetime.

Energy consumption: Fig. 10 depicts the effect of relocation on energy consumption, measured in terms of the average energy per packet. The results show that relocating the gateway achieves energy savings of approximately 10% compared to the baseline approach. This is expected since moving the gateway closer to heavily loaded nodes reduces the transmission power of these sensor nodes and in many cases decreases the number of hops that data packets travels. Such energy savings are translated also to increased node lifetime as seen in Fig. 11. Such boost in node's average lifetime becomes even more significant for larger networks.

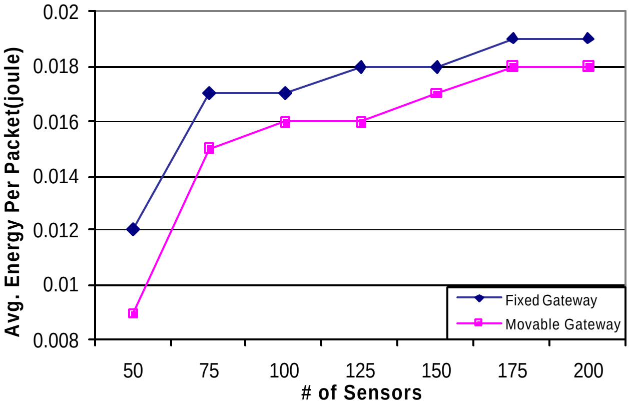


Fig. 10: Average energy consumed per packet for different number of sensors.

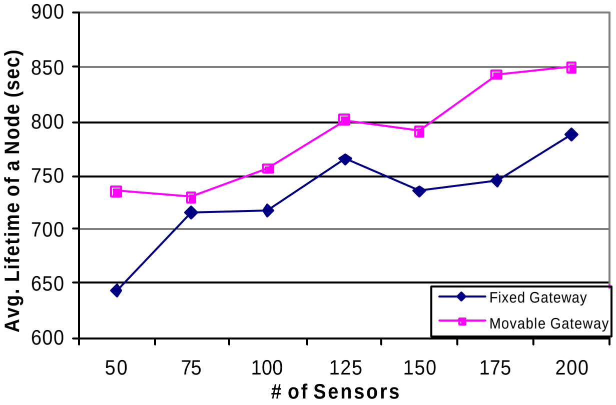


Fig. 11: Average lifetime of a node for different number of sensors.

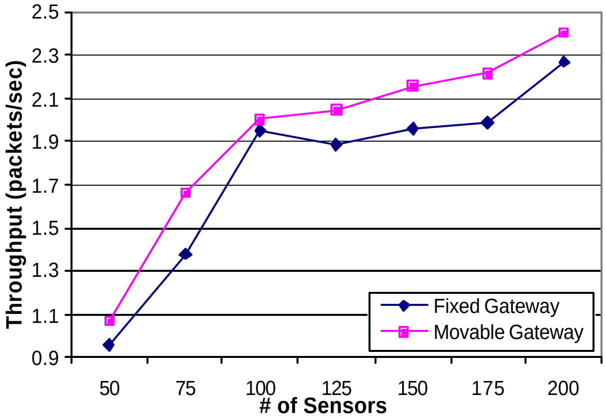


Fig. 12: Network throughput for different number of sensors.

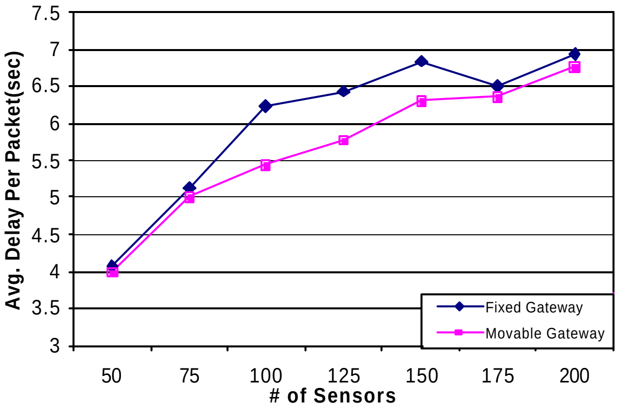


Fig. 13: Average delay per packet for different number of sensors.

Throughput: Gateway relocation also positively impacts the network throughput. As seen in Fig. 12, our approach achieves at least 10% increase in packet throughput. Since our relocation methodology extends the lifetime of nodes as confirmed in Fig. 11, nodes generate/relay more packets. In addition, the gateway often relocates very close to heavily loaded hops decreasing the number of hops traveled by packets. Traveling less number of hops will decrease the probability of packets being dropped and hence more packets will reach the gateway. It is worth noting that we have observed no packets loss due to the movement of the gateway.

Delay: Since the gateway does not always move very close to just one node, the number of hops traveled by most traffic is not consistently reduced and sometimes the packet delay does not improve by the gateway relocation. Our experiments have confirmed such observation, indicating that the gateway relocation yields a slight decrease in average delay experienced by a packet (Fig. 13). The low gain in average packet delay is very much expected given our focus on energy consumption.

4.2.2 Relocation under Constrained Traffic

The performance results in this subsection are for relocating the gateway when time-constrained traffic is being routed. In addition to the metrics considered in section 4.4.1, the deadline miss rate is observed given its importance to real-time traffic. In this case, using the proposed algorithm in Fig. 8 the gateway is relocated 16 times and traveled on average a distance of 28.16m per relocation.

Delay and Timeliness: When qualifying the impact on the average delay and miss rate for real-time packets we have observed that the relocation approach significantly lowers the average delay per real-time packet (See Fig. 14) and provides at least 25% decrease in the deadline miss rate as shown in Fig. 15. This is due to the decreased number of hops and queuing delay for the data coming from highly loaded areas when gateway goes closer to those areas. The decrease in the miss rate is even more significant when the number of sensors is increased, suggesting the positive effect on network scalability.

Energy Consumption: In order to capture the effect of relocation on energy consumption, we have looked at the average energy per packet. The results, depicted in Fig. 16, show that our approach does bring very negligible energy overhead to the network. Although either some of the last hop nodes increase their transmission range or routes are extended through additional forwarders, the potential energy overhead due to such adjustments is compensated through the energy gain when the gateway is relocated towards loaded nodes. Such compensation is more effective when the traffic load is high, demonstrating the scalability of our approach.

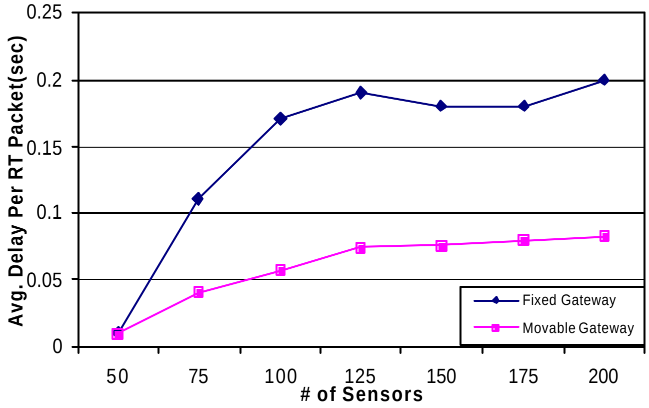


Fig. 14: Average delay per real-time packets for different number of sensors.

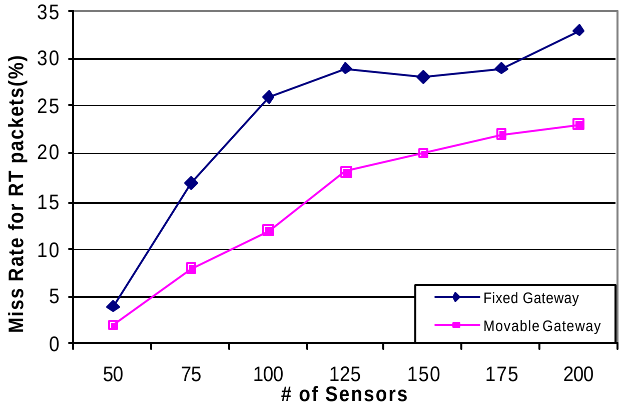


Fig. 15: Deadline miss rate for RT packets for different number of sensors.

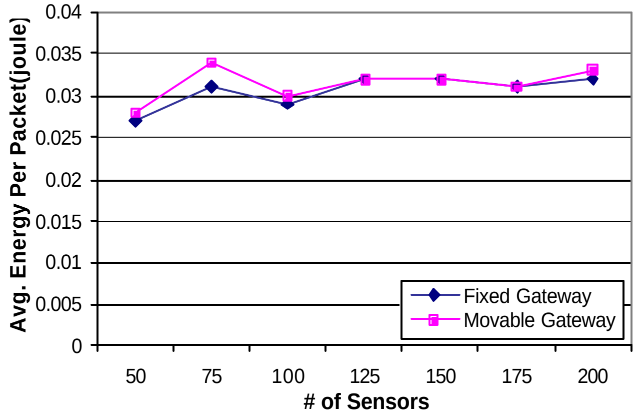


Fig. 16: Average energy/packet consumed for different number of sensors.

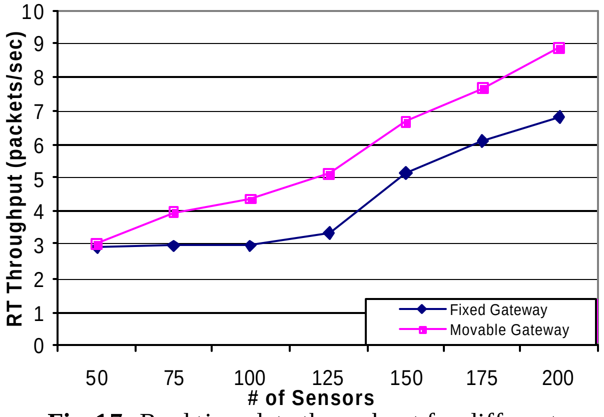


Fig. 17: Real-time data throughput for different number of sensors.

Throughput: In the experiments, we have observed that relocating the gateway increases real-time data throughput by about 20% compared to the setup with a stationary gateway (Fig. 17). Such performance gain is because moving the gateway towards heavily loaded nodes and splitting the traffic will help in boosting the number of admitted flows for real-time relaying and lower packet drops. We again note we have observed no packets loss during the movement of the gateway.

5. Conclusion

In this paper, we have investigated the performance advantage of relocating the sink (gateway) node of wireless sensor networks in response to changes in the network state and traffic pattern. Three main issues have been identified related to when the gateway should be relocated, where it would be best positioned and how the network operation is managed during the gateway's move. The paper has considered both constrained and unconstrained network traffic. For unconstrained data routing, energy metrics are mainly considered to motivate and justify the advantage of gateway's relocation. Our approach checks the traffic density of the nodes that are one-hop away from the gateway and their proximity to the gateway. Once the total transmission power for such nodes is guaranteed to be reduced more than a certain threshold and the overhead of moving the gateway is tolerable, the gateway starts to move to the new position. Simulation results have shown that such repositioning of the gateway increases the average lifetime of the nodes by decreasing the average energy consumed per packet. Moreover, the network throughput is positively impacted.

For delay-constrained traffic, our approach periodically checks the deadline miss rate for real-time packets and triggers a relocation stimulus for the gateway if the miss rate exceeds a certain threshold. In order to designate a new gateway location, our approach finds the node that routes the largest number of real-time packets and checks whether moving to that location of this node or close to it affects the current routes or not. Relocation is performed only if the new location will decrease the end-to-end delay for most real-time packets. Simulation results have demonstrated the effectiveness of our approach showing at least 20% improvement in the deadline miss rate and real-time data throughput when compared to a fixed gateway model. Moreover, the simulation has indicated that our approach boosts these performance metrics without major negative impact on the consumed energy.

References

[1] I. F. Akyildiz et al., "Wireless sensor networks: a survey", *Computer Networks*, Vol. 38, pp. 393-422, 2002.

[2] D. Estrin, et al., "Next Century Challenges: Scalable Coordination in Sensor Networks," in the *Proceedings of the Fifth Annual International Conference on Mobile Computing and Networks (MobiCOM '99),* Seattle, Washington, August 1999.

[3] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, Vol. 43, No 5, pp. 51 – 58, May 2000.

[4] K. Sohrabi, et al., "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, Vol. 7, No. 5, pp. 16-27, October 2000.

[5] R. Min, et al., "Low Power Wireless Sensor Networks", in the *Proceedings of Internation Conference on VLSI Design,* Bangalore, India, January 2001.

[6] J.M. Rabaey, et al., "PicoRadio supports ad hoc ultra low power wireless networking," *IEEE Computer,* Vol. 33, pp. 42-48, July 2000.

[7] W. Rabiner Heinzelman et al., "Energy-Efficient Communication Protocols for Wireless Microsensor Networks," in the *Proceedings of Hawaii International Conference on System Sciences (HICSS '00),* January 2000.

[8] W. Heinzelman et al., "Adaptive protocols for information dissemination in wireless sensor networks," in the *Proceedings of 5th Annual ACM/IEEE MobiCom'99,* Seattle, WA, August 1999.

[9] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks", in the *Proceedings of the 6th Annual ACM/IEEE MobiCom'00,* Boston, MA, August 2000.

[10] R. Shah and J. Rabaey, "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks", in the *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC),* Orlando, FL, March 2002.

[11] D. Braginsky and D. Estrin, "Rumor Routing Algorithm for Sensor Networks," in the *Proceedings of the First Workshop on Sensor Networks and Applications (WSNA),* Atlanta, GA, October 2002.

[12] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," in *Elsevier Journal of Ad Hoc Networks* (to appear).

[13] T. He et al., "SPEED: A stateless protocol for real-time communication in sensor networks," in the *Proceedings of International Conference on Distributed Computing Systems*, Providence, RI, May 2003.

[14] K. Akkaya and M. Younis, "Energy-aware Routing of Delay-constrained Data in Wireless Sensor Networks," *Journal of Communication Systems*, special issue on QoS support and service differentiation in wireless networks, (to appear).

[15] S. Meguerdichian, et al., "Localized Algorithms in Wireless Ad hoc Networks: Location Discovery and Sensor Exposure," in the *Proceedings of the IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC 2001)*, October 2001.

[16] R. Mathew and M. Younis, "Energy-Efficient Bootstrapping Protocol for Sensor Network," in the *Proceedings of the International Conference on Wireless Networks (ICWN'02)*, Las Vegas, Nevada, June 2003.

[17] "Data sheet for the Acoustic Ballistic Module", SenTech Inc., http://www.sentech-acoustic.com/

[18] M. Younis, M. Youssef, K. Arisha, "Energy-Aware Routing in Cluster-Based Sensor Networks", in the *Proceedings of the 10th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS2002)*, Fort Worth, Texas, October 2002.

[19] Y. Yu, D. Estrin, and R. Govindan, "Geographical and Energy-Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks," *UCLA Computer Science Department Technical Report, UCLA-CSD TR-01-0023*, May 2001.

[20] V. Rodoplu and T. H. Meng, "Minimum energy mobile wireless networks," *IEEE Journal on Selected Areas Comunication*, Vol. 17, no. 8, pp. 1333-1344, August 1999.

[21] B. Chen, et al., "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks", in the *Proceedings of the 7th ACM Mobile Computing and Communication (MobiCom 2001)*, Rome, Italy, July 2001.

[22] Y. Xu and J. Heidemann and D. Estrin, "Geography-informed Energy Conservation for Ad Hoc Routing," in the *Proceedings of the 7th ACM Mobile Computing and Communication (MobiCom 2001)*, Rome, Italy, July 2001.

[23] R. Rodrigues et al., "Optimal Base Station Placement and Fixed Channel Assignment Applied to Wireless Local Area Network Projects, " in the *Proceedings of IEEE International Conference on Networks*, Australia, 1999.

[24] T. Fruhwirth and P. Brisset, "Optimal Placement of Base Stations in Wireless Indoor Telecommunication," in *Lecture Notes in Computer Science*, Volume 1520, 1998.

[25] M. Ahmed et al., "Positioning Range Extension Gateways in Mobile Ad Hoc Wireless Networks to Improve Connectivity and Throughput," in the *Proceedings of IEEE Military Communications Conference (MILCOM 2001)*, Washington, D.C., October 2001.

[26] M. Ahmed et al., "Trajectory Control of Mobile Gateways for Range Extension in Ad hoc Networks," *Computer Networks*, Vol. 39, No. 6, pp. 809- 825, August 2002.

[27] A. K. Parekh and G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case", *IEEE Trans. on Networking*, vol. 1, no. 3, pp. 344-357, June 1993.

[28] A. Demers et al., "Analysis and simulation of a fair queuing algorithm" in *Journal of Internetworking Research and Experience*, pp 3-26, October 1990.

[29] A. K. Parekh and G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks Services Networks: The Multiple Node Case", in the *Proceedings of IEEE INFOCOM 1993*.

[30] H.Yang and B.Sikdar,"A Protocol for Tracking Mobile Targets using Sensor Networks," in the *Proceedings of the 1st IEEE Workshop on Sensor Networks Protocols and Applications (SNPA 2003)*, Anchorage, AK, June 2003.

[31] K. Danilidis et al., "Real-time Tracking of Moving Objects with an Active Camera," Real-Time Imaging Journal, Vol. 4, No.1 pp.3-20. February 1998.

[32] J.B. Andresen et al., "Propagation Measurements and Models for Wireless Communications Channels," *IEEE Communications Magazine*, Vol. 33, No. 1, January 1995.

[33] A. Chandrakasan et al., "Power Aware Wireless Microsensor Systems", Keynote Paper ESSCIRC, Italy, 2002.

[34] E. Shih et al., "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," *Proc. of 7th Intl. Conf. on Mobile Computing and Networking*, 2001.

[35] K. Akkaya and M. Younis, "Energy-aware routing to a mobile gateway in wireless sensor networks," in the *Proceedings of IEEE Globecom Ad hoc and sensor networks workshop, 2004* (to appear).

[36] M. Bhardwaj, T. Garnett, and A. Chandrakasan, "Upper Bounds on the Lifetime of Sensor Networks," in *Proceedings of the IEEE International Conference on Communications*, 2001.

[37] A. Sinha and A. Chandrakasan, "Energy Aware Software", in *Proceedings of the 13th International Conference on VLSI Design*, pp. 50-55, Calcutta, India. January 2000.

[38] "Calculating the area and centroid of a polygon," http://astronomy.swin.edu.au/~pbourke/geometry/polyarea/

Appendix A: Sensor's Energy Consumption Model

A typical sensor node consists mainly of a sensing circuit for signal conditioning and conversion, digital signal processor, and radio links [36]. The following summarizes the energy-consumption models for each sensor component.

Communication Energy Dissipation

We use the model of [7][36]. The key energy parameters for communication in this model are the energy/bit consumed by the transmitter electronics (a_{11}), energy dissipated in the transmit op-amp (a_2), and energy/bit consumed by the receiver electronics (a_{12}). Assuming a $1/d^n$ path loss, the energy consumed is:

$$E_{tx} = (a_{11} + a_2 d^n) * r \ and \ E_r = a_{12} * r$$

where E_{tx} is the energy to send *r* bits and E_r is the energy consumed to receive *r* bits. Table 1 summarizes the meaning of each term and its typical value.

Table 1: Parameters for the communication energy model

Term	Meaning
a_{11}, a_{12}	Energy dissipated in transmitter and receiver electronics per bit (Taken to be 50 nJ/bit).
a_2	Energy dissipated in transmitter amplifier (Taken = 100 pJ/bit/m^2).
r	Number of bits in the message.
d	Distance that the message traverses.

Computation Energy Dissipation

We assume the leakage current model of [37]. The model depends on the total capacitance switched by the program and the number of cycles the program takes. We used parameter values similar to those in [37].

Sensing Energy Dissipation

We assume that the energy needed to sense one bit is a constant (a_3) so that the total energy dissipated in sensing *r* bits is [36]: $E_{sensin\,g} = a_3 * r$

For the Ballistic Audio Module sensor [17], the energy dissipated for sensing a bit is approximately equal to the energy dissipated in receiving a bit. Therefore, the parameter a_3 is taken equal to a_{12}.

Appendix B: Computing the centroid of *N* points

The centroid or center of gravity of a polygon can be calculated as follows [38]:

Assume we have *N* locations (x_i, y_i), i=0 to *N-1*. These locations can form a closed polygon if we assume that the last location x_N, y_N is same as x_0, y_0, i.e. only *N-1* distinct sensors. Let c_x be the *X* coordinate and c_y be the *Y* coordinate of the centroid of the polygon respectively. Then

$$c_x = \frac{1}{6A}\sum_{i=0}^{N-1}(x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \text{ and } c_y = \frac{1}{6A}\sum_{i=0}^{N-1}(y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i)$$

where $A = \frac{1}{2}\sum_{i=0}^{N-1}(x_i y_{i+1} - x_{i+1} y_i)$ is the area of the polygon.

In the experiments, we have picked the *X* and *Y* coordinates of the sensors that relay the highest traffic and adjusted them based on each sensor's traffic density. The number of sensors relaying highest traffic is picked up to the median of *PT* (the list that contains the packet traffic of sensors in ascending order) for simulations as explained in section 4.2.1.