MOBILITY AWARE ROUTING WITH PARTIAL ROUTE PRESERVATION IN WIRELESS SENSOR NETWORKS

Sanat Sarangi, Subrat Kar
Bharti School of Telecommunication Technology
And Management, IIT Delhi
New Delhi 110016 India
sanat.sarangi@gmail.com, subrat@ee.iitd.ac.in

ABSTRACT

We propose a novel reactive routing algorithm called PARTROUTE which achieves energy efficient routing in sensor networks with stationary and mobile nodes. PARTROUTE uses coarse mobility awareness in each node to reduce the overheads of reactive routing by intelligently re-using parts of the created on-demand routes beyond their standard lifetime. We show that the responsiveness of the network improves while the overall network traffic is reduced. The effectiveness of PARTROUTE with a large percentage of mobile nodes is also shown.

Keywords: wireless sensor network, routing protocol, PARTROUTE, mobility aware, AODV, AODV Jr

1 INTRODUCTION

A typical sensor network has a single gateway and a set of connected nodes that participate in sensing and forwarding application defined events to the gateway. Some of the key driving factors in designing routing algorithms for sensor networks are (a) size of the network (the number of nodes), (b) limited node energy, (c) deployment topology and (d) provision for node mobility. Traditionally, there have been two approaches to routing– proactive and reactive. In proactive routing, routes can be computed and updated beforehand so that data is sent through the best possible route. In reactive routing, routes are created on-demand. The diversity of sensor network applications have driven the evolution of classes of application appropriate routing algorithms which can be cluster-based [1], hierarchical [2], robust [3], energy-aware [4], geographic [5], query based [6] or swarm-behaviour-based [7].

When some nodes in the network are mobile, reactive routing becomes necessary. Dynamic Source Routing (DSR) [8] and Ad-hoc On-demand Distance Vector Routing (AODV) [9] are popular reactive algorithms developed for mobile ad-hoc networks (MANETs).

2 MOTIVATION

Most reactive algorithms assume that the nodes in the network are mobile. However, mobile nodes are not explicitly aware of their own mobility. Recently, in [10], PH-MA-AODV and Agg-AODV have been proposed that use mobility awareness to improve the performance of reactive algorithms with high-speed MANETs. In PH-MA-AODV a highly mobile node discards a route-request to prevent itself from participating in route formation. In Agg-AODV the route-request accumulates the degree of mobility of each intermediate node and the destination chooses the route with minimum mobility. However, both the algorithms assume that the intermediate nodes can precisely measure their speed of movement. In [11], prediction of broken links is used to trigger routing updates before links break.

Mobility requirements in most sensor networks are different from those of MANETs. The emphasis in sensor networks is on minimizing energy consumption with provision for (often low) mobility within a stationary mesh of nodes where all nodes talk to a central gateway. We propose a routing algorithm PARTROUTE for memory-constrained and energy-sensitive sensor networks with stationary and mobile nodes where each node is explicitly aware of its mobility status at any point of time and this knowledge is used to intelligently preserve part(s) of the route(s) formed during reactive routing beyond the standard route lifetime. PARTROUTE dynamically adapts itself to changing network conditions and helps improve event delivery while reducing both (a) the overall network traffic and (b) latency of route formation (which is critical) for mobile nodes.

With rapid strides in Micro Electro-Mechanical Systems (MEMS), coarse mobility (moving or stationary) awareness can now be easily added to...
sensor nodes with the help of low-cost accelerometers. Accelerometers have been used to detect gait postures like walking, standing, lying [12] [13]. Such information in the context of the behavioural pattern of the subject can be used to know whether a node is (and to predict if is likely to remain) stationary or mobile. In a sensor network containing such semi-mobile nodes, any semi-mobile node that finds itself stationary could temporarily behave like any other (always) stationary node. PARTROUTE is designed to effectively use this behaviour to significantly improve routing performance in such networks.

The rest of the paper is divided into four sections. Section III describes the algorithm. In Section IV we evaluate the performance of PARTROUTE analytically and compare it with the simulation results given in Section V. PARTROUTE is also compared with AODVjr [14] to show the improved performance under different conditions.

3 THE PARTROUTE ROUTING ALGORITHM

In PARTROUTE, we consider a sensor network with \( N \) nodes \( v_1..v_N \) having a set of stationary nodes \( V_S \) and a set of mobile nodes \( V_M \). \(|V_S| + |V_M| = N\) and \( V_S \cap V_M = \emptyset \). Any node \( v_i \) belongs to only one of the sets and adheres to the policies laid out for that set. \( L(v_i, v_j) \) is a link between \( v_i \) and \( v_j \), \( i \neq j \). In PARTROUTE we have two kinds of links—static and dynamic. If two nodes \( v_i, v_j \in V_S \), they are permitted to form a static or a dynamic link. In all other cases the link is dynamic. Links are created and updated only on reception of packets and a table is used to maintain link information. A route between \( v_i \) and \( v_j \) with intermediate node \( v_k \) is denoted by \( P(v_i, v_k, v_j), i \neq j \neq k \). A dynamic link has a lifetime of \( \text{lifetime}_{\text{dynamic}} \) and for a static link it is \( \text{lifetime}_{\text{static}} \). The value for \( \text{lifetime}_{\text{static}} \) is application specific and is further discussed in Sections 4 and 5. It depends on factors like deployment area, number of nodes, mobility behaviour of nodes and their event generation rates.

AODV is an established reactive routing algorithm for MANETs and AODVjr is a successor to AODV which achieves performance comparable to AODV without overheads such as sequence numbers, gratuitous RREP, hop count, hello messages, RERR and precursor lists from AODV. AODVjr is therefore more suitable for memory and energy constrained sensor networks. We compare PARTROUTE with AODVjr.

PARTROUTE uses two kinds of route-requests: M-ROUTE-REQUEST and S-ROUTE-REQUEST for route creation as discussed in Section 3.4 and Section 3.5 respectively. M-ROUTE-REQUEST is processed by \( v_i, v_j \in V_S \cup V_M \) whereas S-ROUTE-REQUEST is processed by \( v_i \) if \( v_j \in V_S \) and discarded if \( v_j \in V_M \). A mobile node uses only M-ROUTE-REQUEST to create a route to the gateway. All stationary nodes in the network except the gateway use S-ROUTE-REQUEST to reach the gateway. The gateway uses M-ROUTE-REQUEST to find a stationary or mobile node in the network since M-ROUTE-REQUEST is accepted and processed by all nodes. We assume that the network has a single gateway and each node in the network is aware of this gateway. The principle can be extended to multiple gateways, without loss of generality.

In PARTROUTE stationary nodes are allowed to behave like mobile nodes under certain conditions. If the duration for which mobile nodes rest (i.e. become stationary) is large when compared to the duration for which they move, they can be allowed to temporarily function as stationary nodes during the stationary phase. The target application determines (a) the use of this role-switching feature and (b) the duration of rest. Role-switching is discussed in Section 3.3.

3.1 Reactive Route Discovery

We use Expanding Ring Search based route discovery [15], an energy efficient method to discover a route to a destination node. The source sends a route-request message with TTL set to 1 and simultaneously starts a timer that expires after twice the node-traversal time. If the destination is one hop away it sends a route-reply message which reaches the source before the timer expires. If a route-reply is not received, the source increases the TTL value, proportionately increases the timeout and repeats the process till either the destination is reached or the source gives up. This limits the propagation extent of the flooded route-request thereby saving energy. The destination sends the route-reply back to the source along the route taken by the first route-request it receives. When a node with a valid route to the destination sends a route-reply on behalf of the destination, the reply is called gratuitous [15].

3.2 Partial Route Preservation

Partial route preservation implies retaining a part of a reactively created route, for a period longer than the standard lifetime for the route. The partial route is called a trace. Each node on a trace is called a representative. A stationary or mobile node which is not a representative is called a non-representative. The gateway is considered to be a representative of itself. In PARTROUTE there can be two kinds of route-replies to a route-request: (a) intermediate-node route-reply that can only be generated by a representative, (b) destination-node route-reply that only the gateway can generate. Henceforth, a node that is allowed to send a route-reply for a route-request is called a reply-node. It may be noted that an intermediate-node route-reply is different from a gratuitous route-reply [15] which is generated by any
node with a route to the destination based on sequence number information already present with the node and that received with the route-request. Sequence numbers are used by neither PARTROUTE nor AODVjr. Moreover, unlike the gratuitous route-reply, a representative does not send any communication to the gateway about the intermediate-node route-reply that it generates.

![Diagram of route formation in PARTROUTE](image_url)

**Figure 1:** Trace formation in PARTROUTE

Fig. 1 illustrates trace formation in the presence of stationary (black) and mobile (white) nodes where mobile node 6 sends a route-request (RREQ) to which the gateway replies. Each stationary node that the route-reply (RREP) encounters on its return path till it reaches a mobile node (node 4 in Fig. 1) becomes a representative candidate. Representative candidates are converted to representatives with create-trace messages discussed in Section 3.3.

Stationary nodes (node 5 in Fig. 1) that occur in a route after a mobile node (node 4) cannot become representatives. The mobile node (node 4) updates the route-reply with this information to notify the stationary nodes that process the route-reply after it about it. Any subsequent request for a route to the gateway by a mobile node can be serviced by the representatives which are closer to the event source than the gateway creating more traces on the stationary nodes. This helps service future route-requests even faster.

Henceforth, we use $S_R$ and $S_U$ to denote the set of representatives and non-representatives respectively. A representative is denoted by $R_i$ and non-representative by $U_i, i \in I'$, where $I'$ denotes the set of positive integers. $S_R \cup S_U = N$ (number of nodes in the network).

### 3.3 Create-trace Messages

A route-reply creates a set of potential nodes called representative candidates which could become representative nodes (Section 3.2). The last of these representative candidates formed by a route-reply (i.e. node 3 in Fig. 1) sends a create-trace (CT) to the gateway and receives an acknowledgement (CA) to become a representative. Since a representative candidate does not know whether it is the last one on the path of the route-reply, the first mobile node that gets the route-reply after a series of representative candidates sends a special message called a create-trace-request (CTR) to the previous node (the last of the representative candidates) asking it to send a create-trace. With no intermediate mobile nodes, the route request originator sends the create-trace if it is the last representative candidate. Create-trace request is not used here. The gateway on receiving a create-trace acknowledges its receipt by sending an acknowledgement to the create-trace originator.

A create-trace message and its acknowledgement convert all dynamic links on the path between create-trace originator and the gateway into static links by updating their link lifetimes to lifetime$_{STATIC}$. The create-trace originator makes a suitable number of retries to send create-trace if it does not receive an acknowledgement before giving up. Acknowledgements may not be received either due to (a) lossy links or (b) role-switching where an intermediate stationary node on the path of create-trace (or its acknowledgement) message suddenly becomes mobile and voluntarily drops the message. Transmission of a create-trace message by a representative for every intermediate-node route-reply that it generates can significantly increase the energy consumption level of the network. We limit this behaviour by defining a variable $CT_{PERIOD}$ such that every time a representative sends a create-trace, it does not send a create-trace for any event received within the next $(\text{lifetime}_{STATIC} - CT_{PERIOD})$ seconds, $0 < CT_{PERIOD} \leq \text{lifetime}_{STATIC}$ to reduce the overheads due to create-trace. Create-trace messages can also become carriers of the node ids of intermediate representative candidates to notify the gateway about their presence.

### 3.4 Route Formation And Route Repair for Stationary Nodes

Stationary nodes only use S-ROUTE-REQUESTs to create routes to the gateway. S-ROUTE-REQUESTs only support destination-node mode operation. So, gateway is the only reply-node. If an S-ROUTE-REQUEST is received by a mobile node, it is simply discarded. So a route created to the gateway with S-ROUTE-REQUEST is guaranteed to have only stationary nodes implying that the trace created in the process is the same as the route. If a stationary node fails to form a route, it performs role switching (Section 3.6) and behaves like a mobile node (Section 3.5).
3.5 Route Formation and Route Repair for Mobile Nodes

Mobile nodes use M-ROUTE-REQUESTs to form routes to the gateway. A mobile node can indicate either intermediate-node or destination-node mode of operation in the M-ROUTE-REQUEST. The intermediate-node mode is used by default and destination-node mode is used under special circumstances as discussed below.

![Diagram](image)

**Figure 2:** M-ROUTE-REQUEST from mobile node 10 extends trace L(1, 2) to P(1, 2, 6)

In the intermediate-node mode, any representative can be a reply-node and send the M-ROUTE-REPLY to an M-ROUTE-REQUEST. A successfully created route between the mobile node and a representative could extend the previous trace terminating at the representative by adding new static links beyond it or create a new trace if the representative is the gateway itself as shown in Fig. 1. Even during trace extension, the create-trace message acknowledgement is generated by the gateway instead of the intermediate node. Existing traces get extended by the recursively applying this principle. In Fig. 2 an M-ROUTE-REPLY generated from node (representative) 2 in response to the M-ROUTE-REQUEST from node 10 forms a route P(1, 2, 6, 10) to the gateway and adds static link L(2, 6) to extend the trace L(1, 2) to P(1, 2, 6).

After creating a route in the intermediate-node mode, if the mobile node discovers that it did not receive an acknowledgement from the gateway for an application level event, it starts sending M-ROUTE-REQUESTs in destination-node mode after a certain number of retries (given by EVENT-RETRY-COUNT) using the intermediate-node mode as determined by the target application. Since gateway is the reply-node for this mode, a route is created between the mobile node and the gateway possibly creating another trace in the process. Once application level communication resumes, the mobile nodes switches back to using intermediate-node mode M-ROUTE-REQUESTs for creating subsequent routes.

Since M-ROUTE-REQUESTs are processed by all nodes they may form shorter traces than S-ROUTE-REQUESTs. This is acceptable to mobile nodes since they would anyway change their location. However, representatives on the trace created in the process would help provide faster responses to other nodes generating M-ROUTE-REQUESTs.

3.6 Role Switching

A stationary node that is unable to find a route to the gateway through other stationary nodes by transmitting a S-ROUTE-REQUEST is allowed to behave like a mobile node (Section 3.5). This allows stationary nodes that have been isolated by deployment or have become isolated during operation possibly due to non-availability of neighbouring stationary nodes to be able to route events through Vs U Vm instead of only Vs. Similarly, a mobile node could become stationary if it remains idle and the target application justifies this conversion after considering the dynamics of the mobile node as discussed in Section 2.

We now explain using Fig. 3 how role-switching by a node participating in route formation affects the behaviour of PARTROUTE after the route is created. Four cases are used to analyze the different circumstances encountered by a mobile node M when it creates a route to the gateway G. A combination of these cases may be used to explain how PARTROUTE handles other situations.

![Diagram](image)

**Figure 3:** Reference diagram for role-switching analysis

**Case 1:** Let us assume a stationary non-representative Uk lies on \(P(M, U_1, U_2, \ldots, U_p, R_1, R_2, \ldots, R_q, G)\). If \(U_k\) switches its role immediately after it processes M-ROUTE-REQUEST, it foregoes its chance to become a representative after the route is formed if \(K = P\) or all nodes between \(U_k\) and \(R_i\) are stationary.

**Case 2:** If \(R_i\) that sent the intermediate-node response to the M-ROUTE-REQUEST becomes mobile then \(\{M, U_1, U_2, \ldots, U_p, R_i\} \subseteq S_U\) and \(\{R_2, \ldots, R_q, G\} \subseteq S_R\). \(R_i\) cannot send intermediate-node route-replies to subsequent M-ROUTE-REQUEST packets since it is not a representative any more.

**Case 3:** If one of the other representatives \(R_i, i \neq 1\) switches its role, its effect is not immediately apparent since M-ROUTE-REQUESTs are fielded by \(R_i\) which comes before \(R_i\). However, now \(\{M, U_1, U_2, \ldots, U_p, R_i\} \subseteq S_U\). If \(R_i\), now a mobile node between representatives \(R_{i-1}\) and \(R_{i+1}\) generates a M-
ROUTE-REQUEST, R_i, could give an intermediate-node response without realizing that its route to the gateway passes through R_n, thus creating a cycle between R_i and R_n. When mobile (or stationary) nodes that use routes with L (R_n, R_i) as an intermediate link fail to receive acknowledgements for events they send to the gateway, they perceive this as a broken link problem and use destination-node M-ROUTE-REQUESTs (or S-ROUTE-REQUESTs) to find new routes to the gateway breaking the above cycle in the process unless the new routes do not use the nodes in the cycle. If nodes in the cycle are not used, the cycle disappears after link lifetimes expire due to absence of communication through the links.

Therefore, rapid role-switching by a mobile node could adversely affect performance of PARTROUTE. So we constrain fidgety mobile nodes to always stay mobile to minimize the overheads due such conditions.

Case 4: If a mobile node U_k switches its role and becomes stationary just after processing the M-ROUTE-REQUEST it either becomes a representative provided K = P (since U_k is adjacent to R_i) or all nodes between U_k and R_i are stationary. Else, U_k remains a stationary non-representative.

4 PERFORMANCE EVALUATION OF PARTROUTE

A large delay in route formation is a bottleneck for reactive algorithms due to their on-demand nature. In the earlier sections, we have shown how traces can successfully help reduce this delay for mobile nodes. In this Section we build an analytical model in order to characterize PARTROUTE’s performance with respect to delay in route formation. The model is used to analyze the simulation results presented in Section 5.

4.1 Assumptions

The nodes are identical with an ideal circular transmission range of radius L. The analysis is carried out with respect events sent to the gateway G by a mobile node M moving within a circular region C_k with radius K as shown in Fig.4. The change in position of M in the interval between sending a route-request and getting back its response is negligible. All transmission, channel and reception conditions are ideal so a route-response is received for every route-request. Radio propagation delay is negligible when compared to the transmission delay at the message source. So, the Hop Traversal Time (denoted by τ, τ > 0) for a message to be received by any node within a radius L of the message transmitter is essentially the same. Mean queueing and processing delays are considered to be a part of τ since they are applicable to every node. We assume a dense uniform deployment of sensor nodes in C_k such that an event transmitted by M can reach the gateway in ⌈x/L⌉ hops through ⌈(x/L)−1⌉ intermediate nodes on a linear path between M and G.

![Figure 4: A circular sensor network deployment for analysis](image)

4.2 Estimating Route Formation Response Time

Suppose at a given instant T, the mobile node M is at a distance x from the G and does not have a route to the gateway. It generates a route-request and expects a route-response. Delay (D_x) experienced by M at a distance x is the duration between the time at which it originates the route-request and the time at which it receives the route-reply. So the cumulative delay experienced by M for all its positions on a ring at distance x and (infinitesimal) width dx from G is $D_x = 2\pi x dx$. Hence, for a given algorithm A, the mean Response Time (RT_A) experienced by M positioned anywhere in C_k is obtained by averaging the cumulative delay over all positions at distance x, $0 \leq x \leq K$ in C_k. This is given as

$$RT_A = \int_0^K D_x 2\pi x dx$$

With the assumptions in Section 4.1, the route request from M is received at the same time by all nodes within a circle of radius L around M. These nodes again broadcast the route-request to repeat the process. In AODVJR only G is allowed to respond to the route request. So, the route request would need at least ⌈x/L⌉ hops to reach G. This is also the worst case situation for PARTROUTE since the gateway would have to field all route-requests in the absence of traces. For this case $D_x = 2\pi [x/L]$. For PARTROUTE, the route-reply may be given by an intermediate node if it is a representative. A representative could be encountered at n hops where
1 ≤ n ≤ [x/L]. Mobility pattern of M is random, trace lifetimes are configuration specific and trace formation depends on earlier route formation attempts and percentage of stationary nodes. Assuming n takes each value within its range with equal probability, the mean value of \( D_r \) is given by
\[
D_r = \frac{2\pi}{\sum_{n=1}^{n} \frac{n}{L}} / [x/L] = \tau([x/L] + 1).
\]

In the best case, all route-requests are fielded at the first hop. So \( D_r = 2\tau \).

Lower the Response Time, lesser is the possibility of the gateway and nodes near it getting choked with route-request packets that could degrade performance in a live scenario. Also, decrease in Response Time improves performance for high speed mobile nodes (further discussed in Section 5.2.2). We also define a figure of merit Trace Index (TI) which is used to compare the effectiveness of PARTROUTE when compared to AODVjr. Smaller the value of TI, better the responsiveness of PARTROUTE.

\[
TI = \frac{RT_{PARTROUTE}}{RT_{AODVjr}}
\]

4.3 Effect of Multiple Event Sources

The performance of PARTROUTE improves as the number of event sources increase since (a) more traces are created and (b) existing traces are preserved for longer durations. More traces help give faster responses to route-requests from mobile nodes thereby controlling the flooding of routing control packets. If a node in the network has a trace to the gateway, for this trace to persist for the entire lifetime of the network it is important that it is refreshed with events at intervals smaller than \( \text{lifetime}_{\text{STATIC}} \).

Deployment parameters especially size of the area and the mean rate of event generation should be used to estimate a suitable value for \( \text{lifetime}_{\text{STATIC}} \) in order to have effective traces.

5 SIMULATION

Discrete event simulation with simulation runs for 1000 seconds is used to evaluate the performance of PARTROUTE. The effect of multiple event sources on PARTROUTE is analyzed. The results obtained for PARTROUTE are compared against AODVjr. Finally, Trace Index discussed in Section 4 is used as a measure of the Response Time of the network for PARTROUTE.

5.1 Conditions

An ideal circular radio model is used for all nodes. There are no packet losses and nodes are always awake. For every event that the node transmits, it receives an acknowledgement from the gateway. Non-receipt of acknowledgement leads to retransmission of the same event instead of the next event. A single gateway, a set (R) of 30 randomly deployed non-event-generating nodes (configured to be static or mobile as per requirement) and one or more event-generating mobile nodes are used to characterize the performance of PARTROUTE. Movement of mobile nodes is random waypoint [8]. Nodes randomly choose a location and a velocity between 1.0m/s to 2.0m/s. After moving to the location with the chosen velocity the next location and velocity are chosen. It may be noted that a stationary node may be either (a) an always stationary node or (b) a role-switching semi-mobile node (discussed in Section 2) which is stationary for the duration of simulation.

![Figure 5: A quarter-circular region with randomly deployed nodes and a gateway](image)

Nodes are deployed in a quarter-circular region \( Q_e \) (a quarter of \( C_K \) in Section 4.1) with radius 300m (Fig. 5). \( Q_e \) is divided into 3 sectors with 10 non-event-generating nodes in each sector. The range of each node (L) is 75m. From Section 4.1, an event generated by a node farthest (i.e. 300m) from the gateway needs 3 intermediate nodes separated from each other by 75m to reach the gateway. However, we use 10 nodes in each sector to (a) sufficiently increase the density of nodes, (b) keep the distribution of nodes across the deployment area homogeneous even with mobility and (c) create a network of reasonable size for analysis. It can be shown that Trace Index for \( C_K \) (in Section 4.2) also holds for each of its four component quarter circles (centered at G with radius K) and hence for \( Q_e \). \( Q_e \) also serves as a reference for analyzing PARTROUTE’s performance in topologies where the gateway is at a corner rather than the centre of the network.

Each non-event-generating nodes when configured to be mobile moves only within its own sector. Event-generating node(s) move in the whole deployment area. The range of each node is kept at
75m to have sufficient overlap between transmission range of nodes. Packet size is kept constant, lifetime$_{DYNAMIC}$ is kept at 15s, CT$_{PERIOD}$ is set to 30s and EVENT-RETRY-COUNT is set to 3. The estimated Hop Traversal Time is 0.1 seconds. An event transmission is attempted by an event-generating mobile node every 6 seconds. If a mobile node in PARTROUTE fails to deliver an event after the retries defined by EVENT-RETRY-COUNT, it uses destination-node M-ROUTE-REQUEST (Section 3.5) till it successfully delivers an event. Henceforth we use PARTROUTE-x to refer to PARTROUTE with the value of lifetime$_{STATIC}$ as x.

We use the term Total Packet Count to denote the total number of (data and routing control) packets transmitted by all nodes. Event Packet Count is used to denote the total number event packets transmitted by all nodes. Event Count refers to the number of events sent by an event source for which it receives an acknowledgement. So, the total number of packets transmitted for every event (TPE) successfully transferred to the gateway is given by

$$TPE = \frac{\text{Total Packet Count}}{\text{Event Count}} \quad (3)$$

5.2 Results

5.2.1 Effect of multiple event sources

From Fig. 6 it is clear that multiple event sources reduce the Response Time of the network as per the discussion in Section 4.3. Also, for a given number of event sources once the created traces get refreshed periodically, any further increase in the value of lifetime$_{STATIC}$ does not significantly improve performance e.g. for a single event source no significant improvement in response time is observed by increasing lifetime$_{STATIC}$ from 150 to 200.

![Figure 6: Variation of Response Time (in seconds) with lifetime$_{STATIC}$ (lifetime$_{STATIC}$ takes values 50, 100, 150, 200 seconds)](image)

5.2.2 Comparison of PARTROUTE with AODVjr

Response Time for PARTROUTE for a single mobile event source does not vary significantly after 150s as shown in Fig. 6 So, PARTROUTE-150 is used for comparison with AODVjr. A certain percentage of the stationary nodes from R are made mobile and the performance of both the algorithms is compared by varying this percentage from 0 to 80. Figs 7, 8 and 9 show variation of the Response Time, Total Packet Count and Event Count for PARTROUTE-150 and AODVjr for 0-80% mobility.

It is observed from Fig. 7 that for stationary nodes, Response Time for PARTROUTE-150 is less than half of that of AODVjr i.e. PARTROUTE-150 is almost twice as responsive as AODVjr. The Response Time remains significantly smaller (60%) even when up to 40% nodes are mobile. As expected, when almost all nodes (80% nodes) are mobile, the two algorithms have similar performance.

![Figure 7: Variation of Response Time (in seconds) for AODVjr and PARTROUTE-150 for different numbers of mobile nodes (0%; all nodes stationary)](image)

For 0% mobility, events in PARTROUTE-150 are sent over the (long) traces created a priori over the stationary nodes whereas in AODVjr, the instantaneously created routes to the gateway are typically shorter. So, marginally higher TPE levels for PARTROUTE-150 when compared to AODVjr (Fig 8) are due to PARTROUTE-150’s relatively higher Event Packet Count. However, as discussed in Section 4.2, when the event-generating mobile node moves at a (relatively higher) speed of 4-5 m/s, PARTROUTE-150 outperforms AODVjr by delivering 3% more events with 21% lower TPE due to its significantly lower Response Time.

![Figure 8: TPE for AODVjr and PARTROUTE-150](image)
With mobility, lengths of traces shorten so TPE decreases. For 40% mobility, TPE for PARTROUTE-150 is 74% of AODVjr—an improvement of 26%. With 80% mobility, due to lack of sufficient stationary nodes PARTROUTE-150 and AODVjr have similar TPE levels. Event Count for PARTROUTE-150 is more than AODVjr for all cases except with 0% mobility where it is almost same. PARTROUTE-150 delivers up to 15% more events than AODVjr with 40% mobility.

To see the effects of semi-mobility (Section 2) with a large stationary period within the duration of the simulation we configure each node in R to keep switching its status between mobile and stationary. The duration for being mobile and stationary are given by two normally distributed variables with mean values of 450s and 50s respectively, each with a standard deviation of 10s. This prevents role-switching from happening synchronously. Each node is initially mobile. PARTROUTE-150 delivers 5% more events, has a 17% lower response time and 8% lower TPE compared to AODVjr.

![Figure 9: Event Count for AODVjr and PARTROUTE-150](image)

A topology in Fig. 5 with random node deployment has been used to characterize the general behaviour of PARTROUTE. However, a typical commonly used configuration would look like Fig. 10 where 50% of the stationary nodes in R (i.e. 15 nodes and the gateway) form a connected grid through which events can be easily routed and the remaining 50% move throughout the grid using the random-waypoint mobility model discussed earlier. There is one event-generating mobile node. Remaining conditions are the same as earlier. Transmission range of nodes and the distance between adjacent nodes on a row or column is 75m. PARTROUTE-150 delivers 12% more events when compared to AODVjr at a 20% lower response time and 20% lower TPE. With appropriate adjustment of node transmission range, the configuration also helps in carrying out localization [16] [17].

![Figure 10: A grid deployment of nodes](image)

5.2.3 Variation in Trace Index

From Section 4.2, theoretical values of Trace Index with L=75 and K=300 are given as 1, 0.66 and 0.32 for worst, mean and best case respectively. Fig.11 shows how the simulation results for Trace Index for PARTROUTE-150 compare against the theoretical worst, mean and best case levels. The lowest value of 0.44 is observed for Trace Index with stationary nodes. With 40% mobility, the observed Trace Index is 0.6 which is better than mean value of 0.66.

![Figure 11: Trace Index Analysis](image)

6 CONCLUSION

We have shown how logically dividing the network into stationary and mobile nodes with partial route (trace) preservation over stationary nodes can help increase the effectiveness of reactive routing in mobility aware sensor networks where the importance of energy-efficiency is paramount. Performance of PARTROUTE under different conditions has been shown to illustrate its applicability to different situations. Mobility characteristics of the nodes comprising the sensor network, nature of event traffic and typical deployment topologies could be used to configure PARTROUTE to bring about significant
improvements on all routing performance parameters. PARTROUTE can also find significant application with Body Sensor Networks where the mobility pattern of nodes can be determined with sensors like accelerometers.

7 REFERENCES


