Published in IET Communications
Received on 2nd July 2008
Revised on 8th June 2009
doi: 10.1049/iet-com.2008.0745

ISSN 1751-8628

Energy-efficient directional routing between partitioned actors in wireless sensor and actor networks

Ka. Selvaradjou   N. Handigol   A.A. Franklin   C.S.R. Murthy
Indian Institute of Technology Madras, Chennai 600036, India
E-mail: selvaraj@cse.iitm.ac.in

Abstract: Actor–actor communication is an important part of the functioning of wireless sensor–actor networks and enables the actor nodes to take coordinated action on a given event. Owing to various reasons such as actor mobility and low actor density, the actor network tends to get partitioned. The authors propose to use the underlying sensor nodes, which are more densely deployed, to heal these partitions. In order to maximise the utilisation of the limited energy available with the sensor nodes, a new routing protocol for actor–actor communication using directional antennas on the actor nodes is proposed. The authors contribution is threefold. First, using simulations they show that the problem of partitioning in the actor networks is significant and propose an architecture with directional antennas on actor nodes and sensor bridges to heal these partitions. Second, they identify the routing problem for this architecture based on a theoretical framework and propose centralised as well as distributed solutions to it. Third, they develop a routing protocol based on the distributed solution and show, using network simulations, that the proposed protocol not only heals the network partitions successfully, but also achieves high throughput and fairness across different flows, in addition to maximising the network lifetime.

1 Introduction

Wireless sensor and actor networks (WSANs) are of tremendous use in today's world and hold a huge promise for future applications [1]. These networks can be an integral part of systems such as battlefield surveillance and attack detection. WSANs consist of sensor and actor nodes. Sensor nodes are low cost, low power and tiny devices with limited sensing, computation and wireless communication capabilities. They sense the surrounding phenomena and communicate the information to the actor nodes. The wireless-enabled actor nodes are capable of acting on the environment such as putting out fire and pumping gas on intruders. These nodes are resource rich and equipped with better processing capabilities, higher transmission powers and longer battery life.

It is common in WSANs that a large number of sensor nodes and a relatively fewer number of actor nodes are deployed in the terrain under monitoring. The lower number of actor nodes are due to the cost associated with them and the deployed actor nodes may be static or mobile depending on the application. Effective sensor-to-actor communication (SAC) and actor-to-actor communication (AAC) are two important problems in WSANs. When events are reported to one or more actor nodes, a coordinated action is needed to meet the real-time deadlines associated with the events. If the deployment of actor nodes is such that the actor network is connected, then AAC is not a serious concern. But if the number of actor nodes deployed in the terrain is not large enough, which is mostly the case, the actor network topology becomes sparse and achieving effective AAC becomes an important problem. As we show later in Section 2, the actor network gets partitioned and as a result no two actor nodes belonging to different partitions can communicate with each other.

In order to heal the partitions in the actor network, we propose to use an architecture that uses intermediate sensor nodes as bridges and directional antennas on actor nodes.

However, the usage of sensor nodes makes energy a critical constraint while routing data using this architecture. We identify the routing problem as that of maximising the amount of AAC data transfered under the constraints imposed by sensor nodes and propose centralised and distributed solutions to the problem. This is a novel aspect of our work and is the main theme of the proposed energy efficient directional routing (EEDR) protocol. To the best of our knowledge, routing for AAC in WSANs has not been studied from this perspective before and our work is the first of its kind in this direction.

EEDR is a novel routing protocol and seeks to achieve high throughput, network lifetime and fairness across flows. The salient features of EEDR are (i) Robustness – EEDR is self-configuring and robust to the dynamics of the WSAN topology. It achieves AAC with minimal disruption even under conditions of high actor node mobility. We emphasise that, while EEDR is designed for AAC in partitioned actor networks, it works perfectly well even when the actor network is fully connected. (ii) Energy awareness – EEDR maximises the network lifetime and the amount of data transfered under the constraints of limited energy of the sensor nodes. (iii) Fairness – Given the limited energy of the bridging sensor nodes, it is important to ensure that this constrained resource is fairly distributed and no actor flow is starved. The algorithm that EEDR operates on ensures high fairness across all the flows.

The rest of the paper is organised as follows. Section 2 describes the motivation behind our work. We highlight the difficulties in providing AAC in a sparse topology of actor nodes in Section 3. Section 4 gives an analytical framework for the routing problem and we propose solutions in the form of centralised and distributed algorithms. The details of our routing protocol are presented in Section 5. The performance of our protocol is evaluated using simulation in Section 6. In Section 7, we summarise the related work. Finally in Section 8, we conclude our work with discussions on future work.

2 Motivation

Consider the battlefield surveillance application [2] of WSANs in which a huge number of sensor nodes are deployed randomly and that the war tanks, robots and/or the soldiers carrying wireless nodes are the actor nodes. In such applications because of the associated cost, the actor nodes cannot be deployed in a large quantity. Thus, it is difficult to maintain the communication coverage among all the actor nodes despite their resource richness. But, it is essential to provide communication between isolated actor nodes for coordinated action. In this section, we justify the need for a routing protocol to heal actor network partitions using sensor-node bridges. Using simulation, we show that the problem of partition in the actor networks is common even for a reasonably large number of actor nodes and can result in a significant number of partitions.

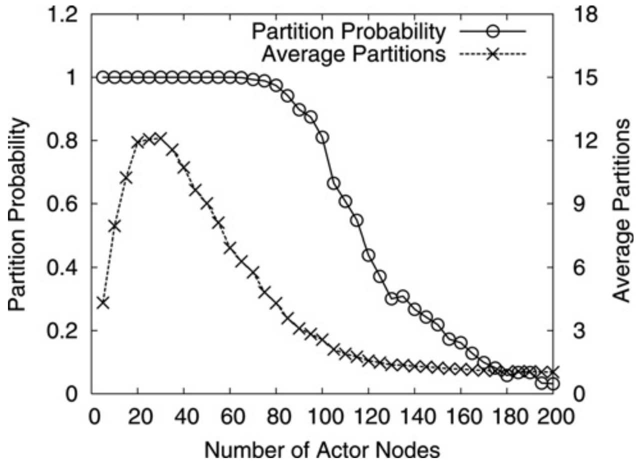

Figure 1 *Partition probability and number of partitions against number of nodes*

We simulated a network of actor nodes in which actor nodes are randomly deployed in a terrain of dimension 1000 m × 1000 m. The communication range of actor nodes is set to 150 m. The network is said to be partitioned if there exist at least two actor nodes that cannot communicate with each other, either directly (single-hop) or indirectly (multi-hop). We varied the number of actor nodes in the terrain to study the connectivity of the resulting graph.

In Fig. 1, we plot the probability that the actor network is partitioned by varying actor node density. From the graph, we can observe that the partition probability remains high and close to 1 for as many as 80 actor nodes in the terrain. Actor nodes are generally sparsely deployed in the terrain and a lower number of nodes almost certainly guarantees a network partition, indicating the need for a solution to heal the partitions.

In the same figure, we plot the average number of partitions that could occur because of random node deployment in the terrain by varying the node density. We observe that the number of partitions is significantly high ($\geq$4) for node-count up to 80 nodes and goes up to as high as 12 partitions. Moreover, because of node mobility, these partitions tend to change and reorganise over time. This necessitates a robust routing protocol to manage communication between these numerous partitions under dynamic conditions.

3 Actor–actor communication on a sparse topology

Fig. 2 shows an overview of a WSAN architecture. The network shown in the figure consists of 11 actor nodes (triangles) and numerous sensor nodes (circles). Actor nodes communicate with each other over a long-range communication channel (shown as zig-zag lines) without interfering with the short-range communication channel (shown as directed arrows). The sparse topology because of the deployment of a small number of actor nodes in a large

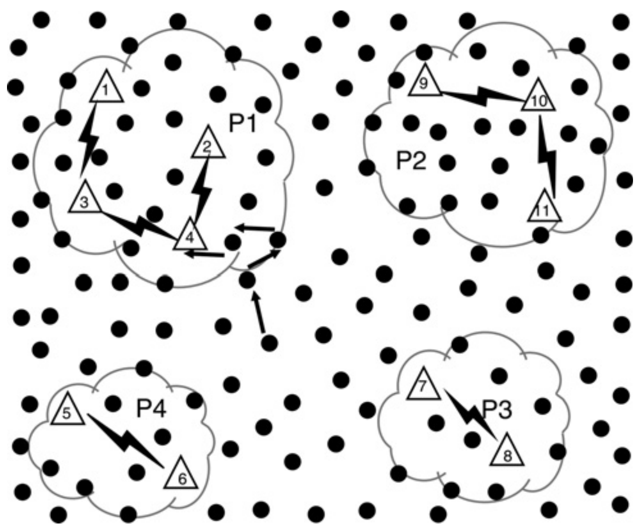

Figure 2 *WSAN architecture overview*

terrain and the frequent mobility of actor nodes results in partitioning of the actor network. As shown in the figure, the actor nodes in partition *P1* cannot directly communicate to any of the actor nodes in other partitions (*P2, P3* or *P4*) and vice versa. However, the sensor nodes, being numerous in the field, can be used to form bridges that heal these partitions. But the sensor nodes, being constrained in terms of energy, memory and computational power, must be intelligently utilised in order to maximise the time for which AAC can continue across the partitions.

When communication between two actor nodes belonging to different partitions needs to be established, actor nodes can switch to a short-range sensor communication channel and thus heal the partition by means of intermediate sensor nodes. But, such a solution would result in an enormous amount of delay in AAC that could not be tolerated by real-time applications. Alternatively, the partitioned actor nodes can switch to a sensor communication channel with long-range transmission to reduce the number of hops and number of intermediate sensor nodes required to act as a bridge between the actor nodes. Although this approach reduces the end-to-end delay in AAC, it comes at the cost of increased collisions with all ongoing sensor channel communications (resulting in the need for retransmissions and adding to sensor node battery drain). Collisions could be avoided by scheduling the sensor nodes to defer their transmissions, but with an omni-directional antenna that would involve increased end-to-end latency for all of the sensor nodes within a 360° long-range radius of the transmitting actor node. In previous work [3], it has been shown that, use of directional antennas on actor nodes can significantly reduce the energy consumption of sensor nodes and losses because of packet collision. We, in this work, design and evaluate the performance of an energy-efficient routing protocol which relies on the underlying medium access control (MAC) layer protocol designed for this heterogeneous architecture. In our proposed protocol, an actor uses a directional antenna to broadcast its data using its maximum power only on a selected cell sector. So, only the nodes that are within the sector will receive the packets, resulting in fewer losses because of packet collision. Moreover, only those sensor nodes that are beyond a certain threshold distance (between 70 and 100% of the long communication range) from the actor node forward the packet, others (closer to the actor) just drop them. Since the number of hops is reduced, this leads to reduced energy and latency.

3.1 Two-layered approach

We address the routing problem with a two-layered approach. In the topology that we consider, the graph of actor nodes is partitioned and consists of a number of partitions with the sensor nodes acting as bridges between these partitions. In our solution, we perform routing at two levels, viz. intra-partition and inter-partition routing.

3.2 Intra-partition routing

The actor nodes within a partition form a connected graph and are capable of communicating over long-range actor channel. Moreover, actor nodes are not energy constrained. Therefore routing protocols similar to those used for mobile ad hoc networks [4, 5] can be used here. Actor nodes within a partition should coordinate appropriately to achieve inter-partition routing.

3.3 Inter-partition routing

Inter-partition routing occurs between actor network partitions that cannot communicate directly over long-range actor channel. Therefore they use the intermediate sensor nodes as bridges for the communication. Sensor nodes, being energy constrained, impose limitations on the amount of data that can be transmitted through them before their energy is drained off. In Section 4.4, we present a distributed algorithm for inter-partition routing that maximises the amount of data that can be transmitted.

4 Theoretical analysis

In a WSAN, actor nodes are energy rich and are capable of communicating a large amount of data compared to the energy-constrained sensor nodes. Therefore, in order to maximise the amount of data transfered in the AAC, the actor nodes should be used to the maximum possible extent and the energy of the sensor nodes should be optimally utilised. This is the intuition behind our formulation of the routing problem in the proposed architecture as a graph theoretic multiple source–destination route scheduling problem.

From the given WSAN, we abstract out a weighted graph $G(V, E)$, where each node $v \in V$ represents a partition of actor nodes and each edge $e \in E$ represents the bridge of sensor nodes that links up two partitions. For the moment,

IET Commun., 2010, Vol. 4, Iss. 1, pp. 102–115

doi: 10.1049/iet-com.2008.0745

we assume that actor nodes within a partition can communicate and coordinate among themselves so that they appear as a single entity to other partitions. In graph G, the weight of each edge e represents the energy of the sensor bridge, which is directly proportional to the maximum number of packets that can be transmitted through the sensor bridge before it breaks. In order to achieve AAC, for every sending and receiving actor node pair, we identify a node pair in G and call them source–destination pairs. Here, the source and the destination are nodes corresponding to the partitions to which the sending and receiving actor nodes belong, respectively. Given a set of source nodes $S = \{s_1, s_2, \ldots, s_k\}$ and the corresponding set of destination nodes $T = \{t_1, t_2, \ldots, t_k\}$, we schedule routes such that network utilisation is maximised.

4.1 Problem definition

Given a weighted graph $G(V, E)$, a set of source nodes $S = \{s_1, s_2, \ldots, s_k\}$, and a set of destination nodes $T = \{t_1, t_2, \ldots, t_k\}$ such that node s_i can have routes only to node t_i, schedule routes between S and T such that the total flow F is maximised. The total flow is defined as

$$F = f_1 + f_2 + \cdots + f_k \tag{1}$$

where f_i is the flow for routes scheduled from s_i to t_i.

4.2 Centralised solution

In this section, we propose a centralised solution for this problem based on second-order cone programming (SOCP) [6] in order to obtain the optimal solution which will be compared with the distributed solution discussed in the following sections. We define the following variables:

$f_i(u, v)$: The number of packets (or bytes) flowing over the edge connecting vertices (u, v) and originating from source s_i for destination t_i.

$c(u, v)$: The capacity of edge (u, v) or equivalently, the maximum number of packets (or bytes) that can flow through the edge connecting vertices u and v.

E_u: The set of neighbouring vertices of vertex u.

We formulate the SOCP problem as follows

$$\text{Maximise} \quad \sum_i \sum_{u \in E_{s_i}} f_i(s_i, u) \tag{2}$$

Subject to the following constraints

$$f_i(u, v) = -f_i(v, u), \forall i, (u, v) \tag{3}$$

$$\sum_{v \in E_u} f_i(u, v) = 0, \forall u \notin S, T \tag{4}$$

$$\sum_{u \in E_{s_i}} f_i(s_i, u) + \sum_{u \in E_{t_i}} f_i(t_i, u) = 0, \forall i \tag{5}$$

$$\sum_i |f_i(u, v)| \le c(u, v), \forall (u, v) \tag{6}$$

The problem is to maximise the objective function (2), which is equal to the overall flow of packets going from every source ($\in S$) to their immediate neighbours. The constraints impose conditions such as each flow has an associated direction (3), intermediate partitions do not generate/consume packets (4), all packets reach their corresponding destinations (5) and edge capacities are not exceeded (6). In summary, the stated problem maximises the overall flow of packets between all the sources in S and their corresponding destinations in T.

This problem can be solved in polynomial time with arbitrary accuracy. Specifically, the authors of [7] show that the long-step path-following algorithm using the Nesterov and Todd (NT) direction has $O(k \log \epsilon^{-1})$ iteration complexity (where ϵ is the duality gap reduction factor and k is the number of second-order cones), which is also the best result for the scaling methods they considered. In general, any centralised algorithm is not pragmatic and it is not suggested to use it in any practical scenarios because of high computational complexity and high communication overhead. Moreover, because of the absence of a central decision-making entity in our actual problem, we want the actor nodes to make routing decisions in an independent and distributed fashion. We propose an approximation distributed algorithm to achieve this.

4.3 Distributed solution – linear programming

The distributed solution consists of each source destination pair (s_i, t_i) determining optimal routes for itself using a linear programming (LP) problem [8]. The LP problem for each source–destination pair can be formulated as follows

$$\text{Maximise} \quad \sum_{u \in E_s} f(s, u) \tag{7}$$

Subject to the following constraints

$$f(u, v) = -f(v, u), \forall (u, v) \tag{8}$$

$$\sum_{v \in E_u} f(u, v) = 0, \forall u \notin S, T \tag{9}$$

$$\sum_{u \in E_s} f(s, u) + \sum_{u \in E_t} f(t, u) = 0 \tag{10}$$

$$f(u, v) \le c(u, v), \forall (u, v) \tag{11}$$

The computational complexity of an LP is $O(n^{3.5}L)$ [8], where n represents the number of variables and L represents the order of space required to input the problem. In our case, the total number of variables is e, where e is the number of edges in graph G. Therefore the time complexity of our solution is $O(e^{3.5}L)$. However, this solution does not address the problem of fairness as it does not provide the order in which the paths are to be chosen for routing. We, therefore, propose an alternative heuristic to perform the computation of near-optimal paths much faster.

4.4 Distributed solution – greedy heuristic

In order to describe the heuristic, we define the following two terms: (i) Path capacity – Given a path between vertices u and v in graph G, the maximum number of packets that can be transmitted over the path from u to v. (ii) Maximum capacity path – Given two vertices u and v in graph G, the path from u to v that has the maximum path capacity.

The pseudo code of the algorithm to determine the maximum capacity path between two vertices is shown in Fig. 3. Here, wt[u, v] refers to the capacity of the edge connecting vertices u and v and cap[v] refers to the path capacity of the maximum capacity path from u to v. The function extract_max(Q) returns the vertex with maximum capacity value, that is, the vertex v such that cap[v] is maximum among all vertices. So, given two vertices (actor node partitions), we say that a vertex a is better than a vertex b if the maximum capacity of the path from the source to a is higher than that of the path to b. The algorithm is based on Dijkstra's Shortest Path Algorithm [9]. Given a source vertex u and a destination vertex v, the pseudo code of the greedy heuristic presented in Fig. 4 determines the route schedule.

Once the optimal routes for every source–destination pair are determined, the AAC results in a global super-imposition of the routes, where each (s_i, t_i) pair tries to utilise all its calculated routes. Whenever there is a conflict in terms of routes sharing edges, the edge capacity (which is the edge weight) gets fairly shared among the competing flows. When the total traffic exceeds the available link capacity, each of the flows gets a lower throughput compared to what it had calculated. In addition, the source–destination pairs utilise the paths (among those that form the calculated routes) in a decreasing order of path capacity (greedy heuristic). These two properties result in a fair distribution of the available energy resources among different source–destination pairs.

The algorithm given in Fig. 3 has the following advantages: (i) It is distributed, enabling each sending–receiving actor node pair to make decisions independent of

```
Max_Cap_Path(G, v):
for all v ∈ V do
  cap[v] = 0
  previous[v] = undefined
end for
cap[source] = ∞ { // Distance from source to source}
Q = copy(Graph) { // All nodes in the graph are unoptimised thus are in Q}
while Q is not empty do
  u = extract_max(Q) { // Remove and return best vertex from Q}
  for all v such that v is neighbour of u do
    alt = min(cap[u] , wt[u, v])
    if alt > cap[v] then
      cap[v] = alt
      previous[v] = u
    end if
  end for
end while
return previous[ ]
```

Figure 3 *Computation of the maximum capacity path between two vertices in a graph*

doi: 10.1049/iet-com.2008.0745

```
G′ = NULL
while u and v are connected do
    P = Max_Cap_Path(G, v)
    Add the edges of P to G′
    for all e such that e is an edge of P in G do
        cap(e) = cap(e) - cap(v)
    end for
end while
```

Figure 4 *Greedy heuristic for computation of max-capacity routes between two vertices in a graph*

other sender–receiver pairs. (ii) It achieves energy efficiency close to optimal value. (iii) It provides fairness across multiple flows.

4.5 Modelling fairness

For a graph with a single source s and a single destination t, the maximum flow that can be achieved is given by the min-cut of the graph. Let us call this maximum flow f^*. Then, the flow that can be achieved in a multiple source–destination problem will be less than or equal to this value, that is, $f_i \le f_i^*$.

For the given problem with k number of flows, we define the fairness metric χ as follows

$$\chi = \frac{(\sum x_i)^2}{(k \times (\sum x_i^2))} \tag{12}$$

where $x_i = f_i/f_i^*$ and $1 \le i \le k$.

The value of χ varies from $1/k$ (completely unfair) to 1 (completely fair). In Section 6, we show by simulation results that our algorithm achieves fairness close to 1.

5 Energy-efficient directional routing

In this section, we discuss the design details of the EEDR protocol. The protocol achieves AAC between pairs of partitioned actor nodes in a WSAN by utilising the underlying sensor node resources in an energy-efficient manner. It is achieved by exploiting the directional antenna capability of actor nodes and by establishing multiple paths from the source to destination actor nodes, which are in different partitions. The protocol follows a combination of the principles from dynamic source routing [4] and ad hoc on-demand distance-vector (AODV) routing [5] protocols.

5.1 Sensor to actor routing

In the automated architecture of WSANs, the actor nodes present in the network act as multiple sinks for the sensor nodes. The sensor nodes sense the environment and report the events on-demand or periodically to the nearby actor nodes. The sensor nodes recognise the presence of actor nodes from the beacon messages, and decide to route the packets to the closest ones, thereby saving energy on multi-hop communication. A simple AODV routing is applicable for this scenario and in our simulations, we considered AODV for SAC.

5.2 Actor-to-actor routing

In WSANs, coordination between actor nodes is essential in order to perform an action on the environment in an optimal and timely manner. Here, the actor nodes, based on the logged events from sensor nodes, decide to exchange messages or commands to other actor nodes in the network. Owing to the fact that network of actor nodes need not be a connected one as discussed in Section 2, two kinds of routing are possible in AAC.

1. *Intra-partition routing:* If the source–destination pair belongs to same partition, then the communication can take place on the long-range actor channel without the involvement of sensor nodes. The communication may result in single-hop or multi-hop depending on the positions of the pair of nodes.

2. *Inter-partition routing:* If the communicating neighbours belong to different partitions, then the long-range actor channel cannot be utilised. It is essential that the intermediate sensor nodes are involved in establishing the communication between source–destination pair. Our proposed EEDR attempts to utilise the long-range sensor channel and the directional antenna capability to establish a route between source–destination pair.

- *Route discovery:* The route discovery process is initiated by the leader node in every partition whenever (i) a flow commences or (ii) the route-timer expires or (iii) a path break is detected. Each actor node maintains a route-cache in its routing table, which consists of the destination node, the actual sequence of sensor nodes in the route and the available energy on the route. All the paths returned by the destination node are stored in the route cache. Accordingly, the routes are selected in the decreasing order of maximum capacity path whenever a path break is detected. New routes are discovered only when the path break is detected on the currently used path and there exists no more paths in route cache. The presence of the route cache helps to minimise the control overhead whenever a new flow originates from the same source. The entries in the cache are cleaned up when the corresponding path breaks occur or after the expiration of route-timer (so that no stale routes are used) and new routes are added to the cache as and when they are discovered.

• *Leader actor election:* To minimise control overhead, the EEDR elects a leader actor within every partition, which is responsible for coordinating the actions of the actor nodes in the process of route discovery. Such a leader election protocol needs to be robust in the events of node mobility. The EEDR protocol uses the leader election algorithm described in [10] as it works well under the recurrent changes in the topology with less message overhead. The algorithm elects a leader based on an extrema-finding concept. Every actor node within a partition which claims to be a candidate to become a leader broadcasts a control packet. The other member nodes receiving the candidate control packet, either defer from sending their willingness to become leader (if they heard from an actor node with node id less than its own id) or express their willingness to become leader by sending a candidate message. The process continues till all the member nodes exchange their willingness, and finally, the node with the smaller node id broadcasts a leader message. As all these control messages are exchanged over the actor channel, the sensor communications are not disturbed. Leader election enables the whole partition to act as a single entity to implement the algorithm presented in the previous section.

5.3 The EEDR mechanism

When a source actor node needs to send data to a destination actor node, it searches in its route-cache and checks if any route is available to the destination. If so, it sends data packets along the route. Otherwise, it initiates the route-discovery process. The actor node first sends the Route Request (RREQ) packet to its partition leader node. If the leader node knows that the destination actor node belongs to the same partition, it forwards the packet to the actor node directly. Otherwise, it forwards the RREQ packet to all actor nodes of the partition and initiates a directional broadcast.

• *Directional broadcast:* The actor nodes divide the space around them into various sectors in which their directional antennas can broadcast. The actor nodes are aware of the various sectors in which their neighbouring actor nodes fall. Upon the receipt of a directional broadcast command, they broadcast the received packets only in the sectors that do not have a neighbour. We assume that the actor nodes have one of the following capabilities: (i) location awareness or (ii) detecting the angle of arrival of a packet [11].

The actor nodes perform this directional broadcast in the sensor-channel using their long-range directional antennas. As illustrated in Fig. 5, an actor node receiving RREQ packet passes the same to the leader actor, which in turn initiates directional broadcast to all its neighbour actor nodes over actor-channel (shown as thick zig-zag lines). The local decisions at each actor node within the partition avoids broadcasts within the shaded region. Thus, the broadcasts happen in an outward fashion (shown as directional antenna lobes) with minimal inter-sector overlap, thereby saving the scarce energy of sensor nodes within the partition.

• *Sensor bridging:* The sensor nodes, upon receipt of the broadcast packet, forward it to the next partition. They do so by checking for the existence of entries corresponding to actor nodes from another partition in their routing table. Upon receipt of packets from the sensor nodes, the actor nodes forward them to their corresponding leader nodes. This process continues till the destination actor node receives the packet.

5.4 Route selection

Once the destination actor node receives the packet, it sets a timer and collects similar packets from the same source but following different routes. Once the timer expires, the destination actor node constructs a graph based on the packets received and selects the routes using the algorithms given in Figs. 3 and 4. It then informs the source node of these routes by reinforcing these routes using Route Reply (RREP) packets. Once, the source node receives a RREP packet, it adds the route to its route cache.

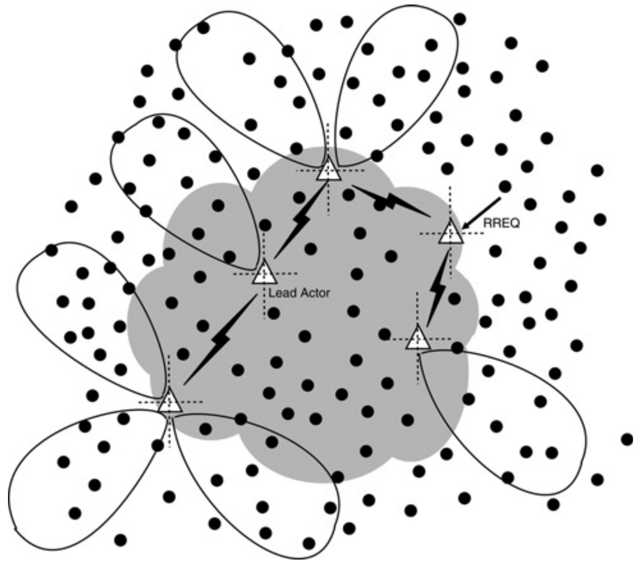

Figure 5 *Directional broadcast*

doi: 10.1049/iet-com.2008.0745

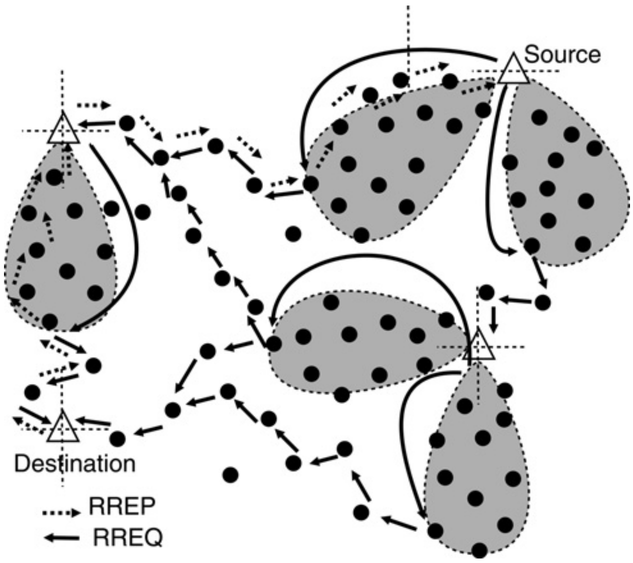

Figure 6 *Route discovery from source to destination*

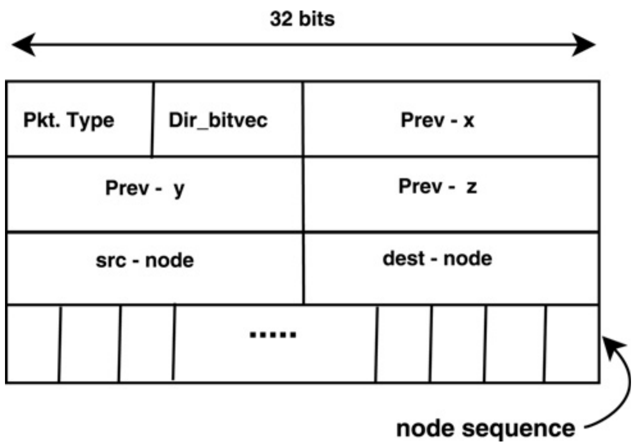

Figure 7 *Packet header structure*

Fig. 6 shows the multi-path routing from the source node after the destination node performs route selection. Owing to the long-range transmission of actor nodes, the directional broadcasts reach sensor nodes farther down the sensor bridge, resulting in lower end-to-end latency and lower energy consumption. These multi-paths are determined by the destination actor node using a greedy heuristic described in Section 4.4. The RREP packets are sent back to the source actor on the reverse acknowledgment path of the selected path and there will be an additional delay because of the limited communication range of the sensor nodes within the destination partition.

6 Simulation and performance evaluation

We implemented and evaluated the performance of EEDR using the ns-2.29 [12] network simulator with energy model given in [13].

6.1 Implementation

Fig. 7 shows the structure of the packet header used by EEDR. It consists of a packet-type field, which indicates the function that the packet is serving. The types of packets include actor ad (beacon broadcast by actor node), leader ad (beacon by leader node within a partition), leader to actor (commands from a leader to member actor nodes in a partition), ADB (directional broadcast by actor nodes), sensor to actor, route reply and actor data.

The other fields in the header include a directional broadcast bit-vector (*Dir_bitvec*) and the location (x, y, z) of the last node that sent/forwarded the packet. In our simulations, we use a 2-D terrain, and hence ignore the z-coordinate. These fields are needed to assist the actor nodes in the directional broadcast phase. *Dir_bitvec* is an array of bits, with each bit representing a particular sector. The MAC layer uses this information to decide which sectors to broadcast the packet in. If the nodes are capable of sensing the direction from which the packets are received, the location fields are not required. The header also consists of a sequence of nodes that the packet has traversed and the energy of each edge. This field is updated by every node as the packet traverses through it.

Depending on the type of the node, the routing table consists of various caches. The caches include Route cache, Leader cache, Neighbour cache and Actor cache.

6.2 Simulation parameters and metrics

Table 1 summarises the various parameters and their settings in our simulation.

The simulation setup consists of sensor nodes randomly distributed in the field. Based on the results presented in Fig. 1, we have simulated a network by deploying 50 actor nodes randomly in the terrain with nine partitions. In order to achieve variation across multiple runs without losing on consistency, we select the source and destination actor nodes for a given flow by first randomly picking two partitions and randomly selecting source and destination nodes among the actor nodes of these partitions for each run. The flows start at different instances of time following Poisson distribution during simulation and the flows carry different amounts of information, with the sizes of each packet kept constant as mentioned in Table 1. The sizes of the flows have been varied between one and five packets. The results presented are averaged over ten runs for each set of parameters.

We use the following metrics for the evaluation of EEDR. All metrics are evaluated as a function of the actor traffic for various number of actor flows.

- *Network lifetime* – We define the death of the network as a point of time at which there exist at least two actor nodes that cannot communicate with each other. The time between the

Table 1 Simulation parameters

simulation area	1000 m $\times$ 1000 m
number of sensor nodes	1000
number of actor nodes	50
number of directional antenna sectors	4, 6
number of actor flows	1, 2, 4, 8
sensor node transmission range	30 m
actor node transmission range	150 m
radio propagation model	two ray ground reflected
packet size	512 bytes (actor), 64 bytes (sensor)
route cache timer	5 s
actor traffic packet inter-arrival time	0.1 to 1 s

beginning of the functioning of the network to the death of the network is the network lifetime.

- *Throughput* – We calculate the average throughput that each flow is able to achieve during the simulation. A high throughput, as close to the packet generation rate as possible is desirable.

- *Route discovery latency* – In the given architecture, path breaks are common, either because of actor node migration or because of low-energy sensor nodes dying down. These path breaks result in discovery of new alternative routes from the source to destination. The time taken for this route discovery is called route discovery latency. A low value of route discovery latency will enhance network throughput and is, therefore, desirable.

- *Fairness* – Fairness has been defined in Section 4.5. A fair protocol would lead to a fairness among the competing flows close to 1.0, whereas an unfair one would lead to a value close to $1/k$, where k is the number of flows.

- *Energy overhead* – We measure the ratio of the energy consumed in transmitting overheads by the sensor nodes to the transmission of actual data packets of actor flows. The overheads include the energy consumption because of route discovery and the message overhead for reporting energy levels of sensor nodes.

We measure the performance of EEDR for various values of the parameters and compare the corresponding results when the directional antennas on the actor nodes operate on four-sector and six-sector setup respectively. In addition, we compare the performance of EEDR with that of AODV protocol [5]. In simulating AODV, we considered the actor having dual channels, viz. long-range actor channel and short-range sensor channel. But, unlike in EEDR protocol, when a source node could not reach the destination over actor channel, it switches to short-range sensor channel. This is due to the fact that AODV lacks the intelligence of directional antenna as in EEDR. In particular, we have not considered long-range sensor channel as an alternate, because it is obvious that it will severely affect the sensor nodes lifetime. The motivation is not to show that EEDR is better than AODV, but to study the performance advantage of EEDR, which inherits the property of AODV, but with intelligence to use directional antenna.

6.3 Static topology

The results of static topology simulations are presented in the Figs. 8–13. In all these figures, the notation EEDR-x, y indicates that the protocol used is EEDR with x sectors and y flows. Similarly, AODV y indicates that the protocol used is AODV with y number of flows.

1. *Network lifetime:* The network lifetime of EEDR is compared with that of AODV by varying the number of simultaneous flows, actor traffic inter-arrival time and directional angle, and the results are presented in Fig. 8. It can be observed from the results that the EEDR achieves as much as 80% improvement in lifetime when compared to AODV irrespective of number of simultaneous flows in the network. Similarly, the use of six-sector directional antenna helps in improving the network lifetime to about 10% more than that of a four-sector antenna. Reduced lifetime because of increase in the number of flows and packet rate can be attributed to the greater number of sensor nodes involved and faster energy consumption.

2. *Throughput:* In Fig. 9, the results of the network throughput by EEDR and AODV are plotted by varying

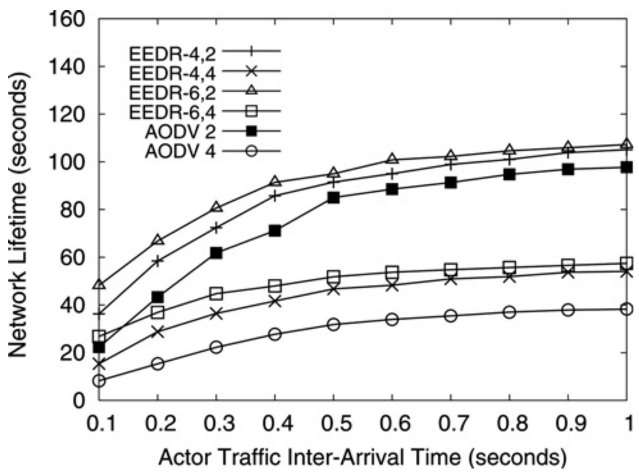

Figure 8 *Network lifetime (static topology) comparison of six- and four-sector EEDR with AODV*

IET Commun., 2010, Vol. 4, Iss. 1, pp. 102–115
doi: 10.1049/iet-com.2008.0745

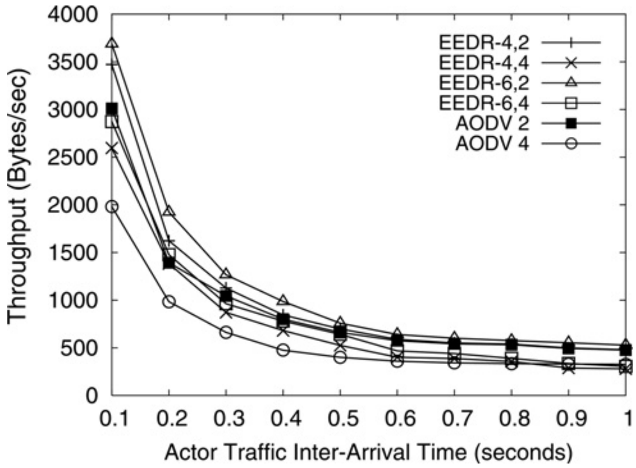

Figure 9 *Throughput (static topology) comparison of six- and four-sector EEDR with AODV*

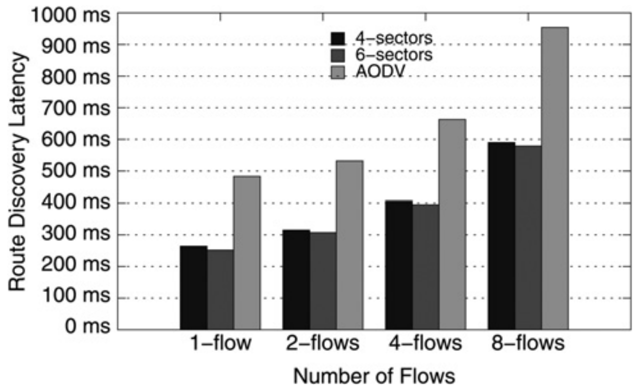

Figure 10 *Route discovery latency (static topology) comparison of six- and four-sector EEDR with AODV*

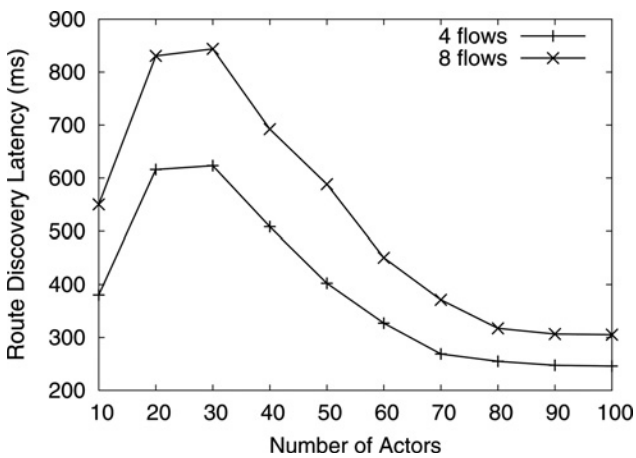

Figure 11 *Route discovery latency of EEDR against number of actors*

number of sectors and number of flows. The decrease in throughput with increase in packet inter-arrival time is entirely due to the fact that the packets are generated at a slow pace such that the maximum capacity of the channel could not be utilised properly. Notably, the results show that the use of narrow bandwidth directional broadcasts enhances the throughput. In general, the throughput of EEDR is significantly higher than that of AODV.

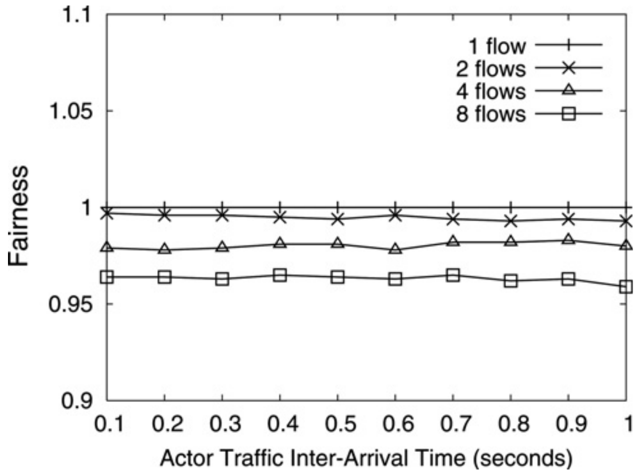

Figure 12 *Fairness (static topology)*

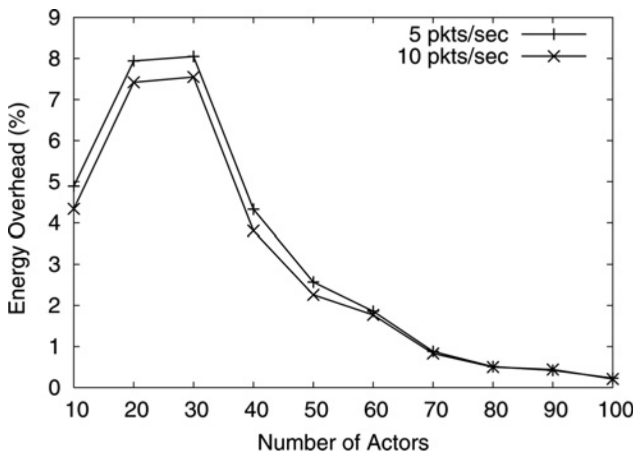

Figure 13 *Energy overhead against number of actors*

3. *Route discovery latency:* Fig. 10 compares the route discovery latency obtained with actor antennas of two different sector angles of 90° (four-sectors) and 60° (six-sectors) with that of AODV. Since EEDR exploits the long-range communication capability of actor nodes, it achieves much lower route discovery latencies compared to that of AODV. The reduced latency in the 60° sector angle case is due to the greater range of the narrower beam [3]. The gain in latency, however, is only marginal because the time required for the route reinforcement (as discussed in Section 5.4) remains the same in both the cases. The results in Fig. 11 show the route discovery latency as a function of varying the number of actor nodes (and thus the partitions) in the network. As can be observed, the latency is predominant when the number of partitions is significant and it is as much as 850 ms for eight flows. As the number

of actor nodes is increased, the latency decreases to a small value. This can be attributed to the reduced number of partitions and hence the intra-partition AAC is predominant.

4. *Fairness:* Fig. 12 shows the variation of fairness (defined in Section 4.5) with actor traffic for various number of flows. We can observe that EEDR achieves very good fairness (>0.95) for as much as eight simultaneous flows.

5. *Energy overhead:* To study the energy overhead because of EEDR protocol, we measured the percentage of overhead as a function of number of actor nodes. The results of Fig. 13 show the energy overhead ratio for two different packet generation rate by actor nodes. The results show that the overhead is marginal and it is as much as 8% only when the number of actor nodes in the network result in maximum number of partitions. The observed overhead is minimal as the number of actor nodes is more in the terrain. This happens because of the reduced involvement of sensor nodes in AAC.

6.4 Dynamic topology

In this set of experiments, we simulate a dynamic topology caused because of actor node mobility and measure the performance of EEDR. We assume that the node movement is caused because of events occurring in the field. We generate the events as a Poisson process with varying arrival rate and uniform distribution in space. When an event occurs, the actor node closest to the event starts moving towards the event with a constant speed. If an actor node is required to act on more than one event, then the actor node chooses to move towards the latest event and 'drops' the other events. Thus, a higher event-rate leads to more frequent actor node movements leading to a more dynamic topology. Table 2 summarises the various parameters and their settings in our dynamic topology simulation. The remaining parameters are the same as those used in the static topology simulations.

1. *Network lifetime:* Fig. 14 shows the variation of the network lifetime with mean inter-event arrival time for a network by varying number of flows. Note that the network lifetime achieved is close to that in a static topology. We observe that the lifetime decreases with increasing number of flows. This is due to increasing contention for the same resource, namely the energy of the underlying network of sensor nodes. Also, for a given number of flows, the network lifetime increases

Table 2 Dynamic topology parameters

number of directional antenna sectors	4
actor traffic packet inter-arrival time	0.4 s
number of flows	1, 2, 4, 8
actor speed	2 m/s
mean event inter-arrival time (s)	10, 20, 30, 40, 50

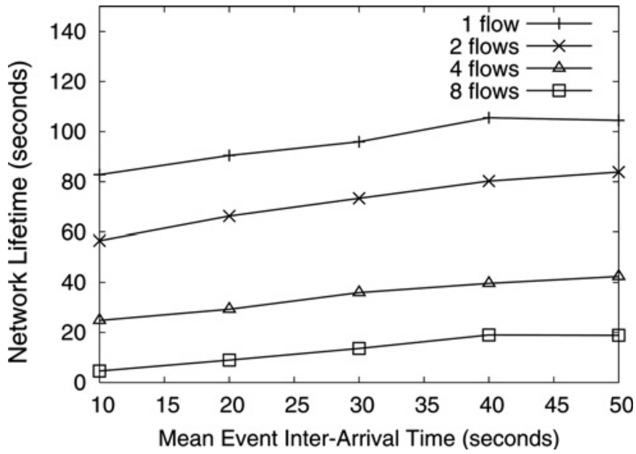

Figure 14 *Network lifetime (dynamic topology)*

with decreasing event rate. This is because of two reasons. First, higher event rate results in a more dynamic actor network topology, which in turn results in more frequent path breaks leading to more aggressive route discovery broadcasts. Second, greater actor node movement results in increased contention for sensor node resources that may cause non-optimal utilisation of the available resource.

2. *Throughput:* Fig. 15 shows the variation of per-flow throughput with mean inter-event arrival time for various number of flows. We observe that EEDR not only maintains connectivity, but also provides good throughput in dynamic conditions. We can also observe that the throughput decreases with increasing number of flows, which is due to increasing contention for the available sensor node energy. For a given number of flows, the throughput increases with decreasing inter-event arrival rate. This is because, lower event-rate results in a more stable network topology. This implies that the routes once discovered can be utilised to a greater extent before new routes need to be used. On the other hand, a dynamic topology leads to frequent path discovery and incomplete utilisation of the discovered routes.

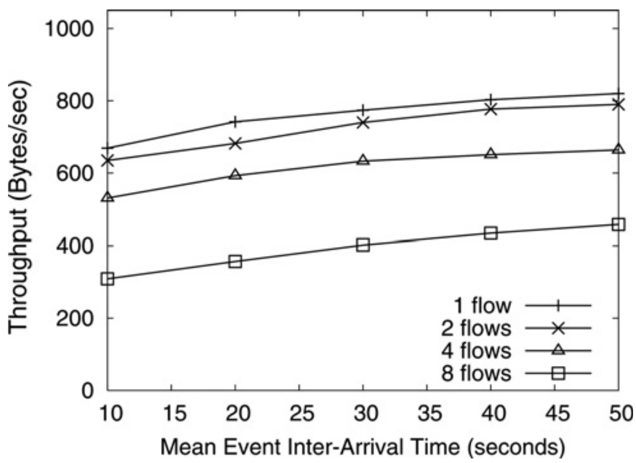

Figure 15 *Throughput (dynamic topology)*

IET Commun., 2010, Vol. 4, Iss. 1, pp. 102–115
doi: 10.1049/iet-com.2008.0745

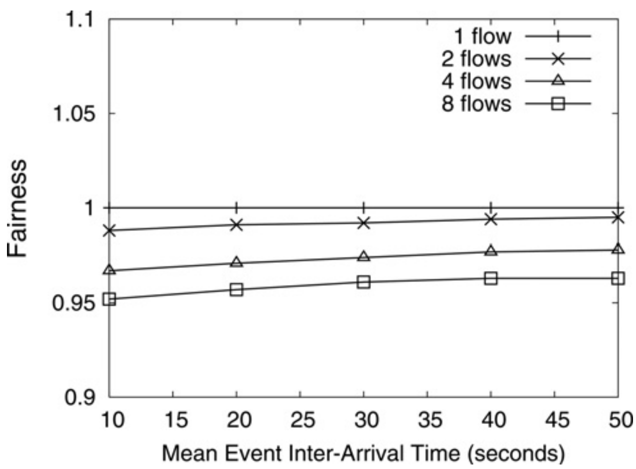

Figure 16 *Fairness (dynamic topology)*

3. *Fairness:* Fig. 16 shows the variation of fairness across flows with mean inter-event arrival time for various number of flows. We can observe that EEDR provides high fairness (close to those obtained for a static topology) even under dynamic network conditions. The fairness decreases slightly with increasing number of flows and remains almost constant for varying event rate.

6.5 Performance of the distributed solution

Fig. 17 shows the comparison of the performance of EEDR with that of the centralised solution (SOCP) (Section 4.2). The centralised solution is obtained by solving the optimisation problem on the network graph obtained as a result of the route discovery process in the WSAN topology simulation discussed in Section 6.3. The optimisation problem is solved using the general algebraic modelling system (GAMS) [14]. The performance of the distributed solution is measured by running EEDR for two different packet rates – 10 and 5 packets/s, in order to study the overhead of route discovery process. A lower

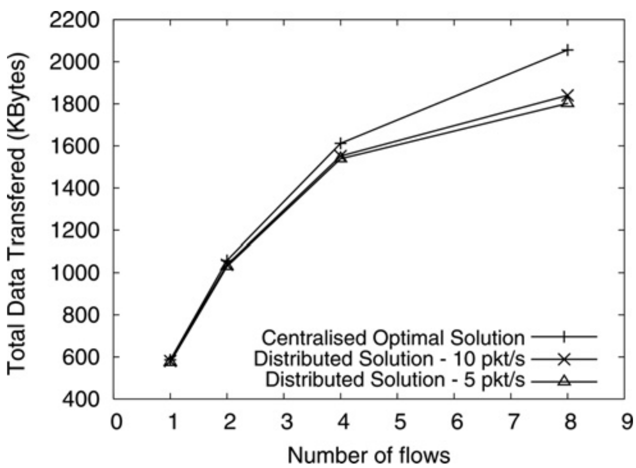

Figure 17 *Comparison of the performance of the distributed and centralised solutions*

packet rate results in a longer network lifetime, hence a larger number of periodic route discoveries.

The figure shows that the achieved throughput in terms of the total amount of data using the distributed heuristic is very close (within 10%) to the optimal values calculated using the centralised solution. It is also to be noted that the difference in the performance of the two solutions is also partially because of the overhead of periodic route discovery, which is ignored in the centralised solution. Also, the difference between the total amount of data transfered for the two packet rates is very low, indicating a very low overhead associated with the route discovery process.

7 Related work

Although energy-efficient communication between sensor and actor nodes is an important problem, effective coordination among multiple actors for collaborative decision to perform coordinated actions is also considered to be important problem. The authors of [15] propose a sensor–actor coordination model based on event-driven partitioning of actors such that sensor nodes are partitioned to different sets associated with one or more actors. In the same work, actor coordination is modelled as a joint optimisation problem. Similarly, energy-efficient routing towards multiple actors in a WSAN is an important problem. Power-speed [16] is a power aware routing protocol that performs energy-efficient and timely reporting of events from sensor nodes to any of the nearest actors. The work follows from the principles of Anycast [17] in which actor nodes are assumed to be connected always and thus the sensor nodes need to put minimal effort in data delivery. However, as we showed in Section 2, such an assumption of connectivity of actors at all times does not hold true always and hence we address the problem of bridging the actor partitions through resource-constrained sensor nodes. Siphon [18] considers the case of diverting the traffic generated by sensor nodes to the physical sink via a set of virtual sinks (static deployment) in case of congestion notification. As in our work, the virtual sinks in [18] are assumed to be resource rich and have dual radio, viz. short range (in-band, typically Mote radio) and far range (out-band, typically IEEE 802.11). In case of network partition in the virtual sink backbone network, it is suggested that the intermediate sensor nodes are used to bridge the gap. However, as the virtual sinks use short-range radio, the resulting end-to-end latency will be enormous which we intend to minimise in our work. In [19], the authors evaluate the performance analysis of the existing routing protocols and claim that dynamic source routing (DSR) is the best suited routing for AAC. The $(RT)^2$ [20] protocol is designed to provide a reliable and real-time delivery of packets between sensor and actors in WSAN. Although the evaluation study in [19, 20] is performed for the case of multiple mobile actors in WSANs, the works do not consider the specific case of partitions in the network of actors which we address in this work.

doi: 10.1049/iet-com.2008.0745

The use of directional antenna at sink node is proposed in [21, 22] in order to extend the lifetime of the sensor nodes in relay zone. The key idea lies in effective scheduling of sensor nodes in relay zone such that nodes wake-up only during the period when sink node's directional beam focuses on them. However, these works do not address the problem of multiple sinks and network partitioning. The work presented in this paper deals with design of an efficient routing protocol that utilises the intermediate sensor nodes in bridging the actor partitions.

8 Conclusions and future work

Actor–actor communication is an important part of the functioning of WSANs and enables the actor nodes to take coordinated action on a given event. We proposed to use the underlying sensor nodes, which are more densely deployed, to heal the actor network partitions. In order to maximise the utilisation of the limited energy available with the sensor nodes, we proposed a new routing protocol for AAC along with long-range directional antennas on the actor nodes.

Our contribution is threefold. First, using simulations we showed that the problem of partitioning in the actor networks is very common. Second, we identified the routing problem for this architecture based on a theoretical framework and proposed centralised as well as distributed solutions to it. Third, we developed a routing protocol based on the distributed solution and showed, using simulations, that our protocol not only heals the network partitions successfully, but also achieves high throughput and fairness across different flows, in addition to maximising the network lifetime.

In future, we plan to study the effect of our routing protocol under a realistic radio and mobility model in which the transit paths of actor nodes are no more obstacle free. Unlike in SAC, AAC demands a reliable transport protocol. Thus, we will extend the work towards designing a reliable and low-energy transport protocol that works well with our proposed communications architecture.

9 Acknowledgment

This work was supported by the Department of Science and Technology, New Delhi, India.

10 References

[1] AKYILDIZ I.F., KASIMOGLU I.: 'Wireless sensor and actor networks: research challenges', *Ad Hoc Netw.*, 2004, **2**, (4), pp. 351–367

[2] BOKAREVA T., HU W., KANHERE S., *ET AL.*: 'Wireless sensor networks for battlefield surveillance'. Proc. Land Warfare Conf. (LWC 2006), October 2006, pp. 1–8

[3] SELVARADJOU K., GOUTHAM B., MURTHY C.S.R.: 'Exploiting resource-rich actors for bridging network partitions in wireless sensor and actor networks'. Proc. Int. Conf. Distributed Computing and Networking (ICDCN), January 2008

[4] JOHNSON D.B., MALTZ D.A.: 'Dynamic source routing in ad hoc wireless networks', in IMIELINSKI T., KORTH H.F. (EDS.): 'Mobile computing' (Kluwer Academic Publishers, 1996), pp. 153–181

[5] PERKINS C.E., ROYER E.M.: 'The ad hoc on-demand distance-vector protocol', in PERKINS C.E. (ED.): 'Ad hoc networking' (Addison-Wesley Longman Publishing Co. Inc., 2001), pp. 173–219

[6] KUO Y.-J., MITTELMANN H.D.: 'Interior point methods for second-order cone programming and OR applications', *Comput. Optim. Appl.*, 2004, **28**, (3), pp. 255–285

[7] TSUCHIYA T.: 'A convergence analysis of the scaling-invariant primal-dual path-following algorithms for second-order cone programming', *Proc. Optim. Methods Softw.*, 1999, **11**/**12**, pp. 141–182

[8] GONZAGA C.C.: 'On the complexity of linear programming', *Resenhas – IME-USP J.*, 1995, **2**, (2), pp. 197–207

[9] CHERKASSKY B.V., GOLDBERG A.V., RADZIK T.: 'Shortest paths algorithms: theory and experimental evaluation'. Proc. Ann. ACM-SIAM Symp. Discrete Algorithms, 1994, pp. 516–525

[10] BOUKERCHE A., ABROUGUI K.: 'An efficient leader election protocol for mobile networks', Proc. ACM Int. Conf. Wirel. Communication and Mobile Computing (IWCMC), 2006, pp. 1129–1134

[11] MAO G., FIDAN B., ANDERSON B.D.O.: 'Wireless sensor network localization techniques', *Comput. Netw.*, 2007, **51**, pp. 2529–2553

[12] 'The Network Simulator NS-2'. Available at: www.isi.edu/nsnam/ns/

[13] ANASTASI G., FALCHI A., PASSARELLA A., CONTI M., GREGORI E.: 'Performance measurements of motes sensor networks'. Proc. Seventh ACM Int. Symp. Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWIM04), 2004, pp. 174–181

[14] BROOK A., KENDRICK D., MEERAUS A.: 'GAMS, a user's guide', *SIGNUM Newslett.*, 1988, **23**, (3–4), pp. 10–11

[15] MELODIA T., POMPILI D., GUNGOR V., AKYILDIZ I.: 'Communication and coordination in wireless sensor and actor networks', *IEEE Trans. Mobile Comput.*, 2007, **6**, pp. 1116–1129

[16] ZHOU Y., NGAI E.C.H., LYU M.R., LIU J.: 'POWER-SPEED: a power-controlled real-time data transport protocol for

doi: 10.1049/iet-com.2008.0745

wireless sensor and actuator networks'. Wireless Communications and Networking Conference, March 2007, pp. 3736–3740

[17] HU W., JHA S., BULUSU N.: 'A communication paradigm for hybrid sensor/actuator networks'. Proc. IEEE Int. Symp. on Personal, Indoor, and Mobile Radio Communications (PIMRC), September 2004, pp. 47–59

[18] WAN C.Y., EISENMAN S.B., CAMPBELL A.T., CROWCROFT J.: 'Siphon: overload traffic management using multi-radio virtual sinks'. Proc. ACM Conference on Embedded Networked Sensor Systems (SenSys), November 2005, pp. 116–129

[19] DINH D.V., VUONG M.D., NGUYEN H.P., NGUYEN H.X.: 'Wireless sensor actor networks and routing performance analysis'. Proc. Int. Workshop on Wireless Ad Hoc Networks (IWWAN 2005), May 2005, pp. 1–6

[20] GUNGOR V.C., ÖZGÜR B., AKAN, AKYILDIZ I.F.: 'A real-time and reliable transport $(RT)^2$ protocol for wireless sensor and actor networks', *IEEE/ACM Trans. Netw.*, 2008, **16**, (2), pp. 359–370

[21] CHO J., LEE J., KWON T., CHOI Y.: 'Directional antenna at sink (DAaS) to prolong network lifetime in wireless sensor networks'. Proc. European Wireless Conf., April 2006, pp. 1–5

[22] MAO S., HOU Y.: 'BeamStar: an edge-based approach to routing in wireless sensor networks', *IEEE Trans. Mobile Comput.*, 2007, **6**, pp. 1284–1296

doi: 10.1049/iet-com.2008.0745