Energy Efficient Directional Routing for Effective Actor-Actor Communication in Wireless Sensor and Actor Networks

Nikhil Handigol, Kandasamy Selvaradjou, A. Antony Franklin, C. Siva Ram Murthy
Indian Institute of Technology Madras
Chennai 600036, India
{nikhilah,selvaraj,antony}@cse.iitm.ernet.in, murthy@iitm.ac.in

Abstract

Actor-actor communication is an important part of the functioning of Wireless Sensor-Actor Networks (WSANs) and enables the actors to take coordinated action on a given event. Due to various reasons such as actor-mobility and low actor-density, the actor network tends to get partitioned. We propose to use the underlying sensor nodes, which are more densely deployed, to heal these partitions. In order to maximize the utilization of the limited energy available with the sensor nodes, we propose a new routing protocol for actor-actor communication using directional antennas on the actor nodes. Using network simulations, we show that our protocol not only heals the network partitions successfully, but also achieves high throughput and fairness across different flows, in addition to maximizing the network lifetime.

I. Introduction

Wireless Sensor-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 *Actors nodes* are capable of acting on the environment such as putting off fire and pumping gas on intruders. These nodes are resource rich and equipped with better processing capabilities, higher transmission powers, and longer battery life.

In WSANs, a large number of sensor nodes are statically deployed and a relatively fewer number of mobile actor nodes are placed randomly. 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. Effective Sensor to Actor Communication (SAC) and Actor to Actor Communication (AAC) are two important problems in WSANs. When the actor network is sparse or the actor nodes are mobile, the actor network gets partitioned and achieving effective AAC becomes an important problem.

In order to heal the partitions in 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 maximizing the amount of AAC data transfered under the constraints imposed by sensor nodes and propose centralized and distributed solutions to the problem. This is a truly 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

- *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.
- *Energy awareness* - EEDR maximizes the network lifetime and the amount of data transfered under the constraints of limited energy of the sensor nodes.
- *Fairness* - The algorithm that EEDR operates on ensures high fairness across all the flows.

The rest of the paper is organized as follows. We highlight the difficulties in providing AAC in a sparse topology of actor nodes in Section II. Section III gives an analytical framework for the routing problem and we propose solutions in the form of centralized and distributed algorithms. The details of our protocol are described in Section IV. The performance of our protocol is evaluated using simulation in Section V. In Section VI, the relevant work in the literature is presented. Finally in Section VII, we conclude our work with discussions on future work.

II. Actor-Actor Communication on a Sparse Topology

Figure 1 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 sensor communication channel (shown as directed arrows). The sparse topology due to the deployment of a small number of actor nodes in a large 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* can not directly communicate to any of the actor nodes in other partitions (*P2, P3, or P4*) and vice versa.

When communication between two actor nodes belonging to different partitions need to be established, actor nodes can heal the partition by using the intermediate sensor nodes by switching either to short-range or long-range sensor communication channel. While the former solution would result in enormous amount of delay in AAC, the latter solution would cause the sensor nodes to suffer severe energy drain as the collision domain would be larger. In a previous work [2], it has been shown that, use of directional antennas on actor nodes can significantly reduce the energy consumption of sensor nodes and losses due to packet collision.

A. Two-Layered Approach

We address the routing problem with a 2-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.

1) 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 [3] [4] can be used here.

2) 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. In Section III-C, we present a distributed algorithm for inter-partition routing that maximizes the amount of data that can be transmitted.

III. 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 maximize the amount of data transfered in AAC, the actor nodes should be used to the maximum possible extent and the energy of the sensor nodes should be optimally utilized. 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, we assume that actor nodes within a partition can communicate and coordinate amongst themselves so that they appear as a single entity to the 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 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 $S = \{s_1, s_2, ..., s_k\}$ and the corresponding set of destination nodes $T = \{t_1, t_2, ..., t_k\}$, we schedule routes such that the network utilization is maximized.

A. Problem Definition

Given a weighted graph $G(V,E)$, a set of source nodes $S = \{s_1, s_2, ..., s_k\}$, and a set of destination nodes $T = \{t_1, t_2, ..., t_k\}$ 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 maximized. The total flow is defined as:

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

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

B. Centralized Solution

We propose a centralized solution for this problem based on Second Order Cone Programming (SOCP) [5]. We define the following variables:

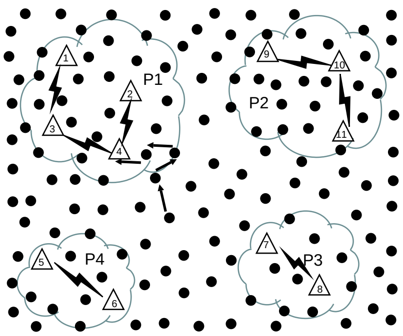

Fig. 1: WSAN architecture overview

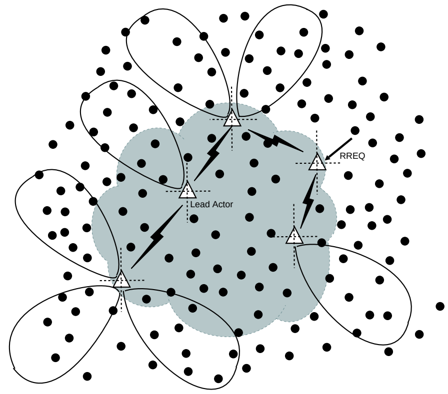

Fig. 2: Directional broadcast

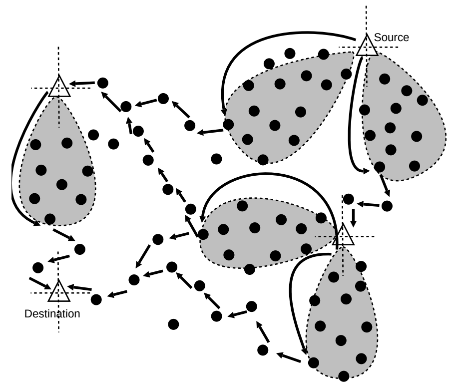

Fig. 3: Multi-path routes from source to destination

$f_i(u, v)$: 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 neighboring vertices of vertex u.

We formulate the SOCP problem as follows.

Maximize

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

Subject to the following constraints:

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

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

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

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

The problem is to maximize the overall flow of packets between all the sources in S and their corresponding destinations in T (Eqn. 2). The constraints impose conditions such as each flow has an associated direction (Eqn. 3), intermediate partitions do not generate/consume packets (Eqn. 4), all packets reach their corresponding destinations (Eqn. 5), and edge capacities are not exceeded (Eqn. 6).

This problem can be solved in polynomial time with arbitrary accuracy. Specifically, the authors of [6] show that the long-step path-following algorithm using the Nesterov and Todd (NT) direction has $O(klog\epsilon^{-1})$ iteration complexity (where ϵ is the duality gap reduction factor and k is the number of second-order cones). However, due to the absence of a central decision making entity in our actual problem, we would want the actor nodes to make routing decisions in an independent and distributed fashion. We propose an approximation distributed algorithm to achieve this.

C. Distributed Solution - Greedy Heuristic

In order to describe the heuristic, we define the following two terms:

- *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.
- *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 Algorithm 1. 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 algorithm is based on Dijkstra's Shortest Path Algorithm [7]. Given a source vertex u and a destination vertex v, the pseudo code of the greedy heuristic presented in Algorithm 2 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 utilize 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 nodes. In addition, the source-destination pairs utilize the paths (among those that form the calculated routes) in a decreasing order of *Path Capacity* (greedy heuristic). These two properties result in

Algorithm 1 Computation of the *Maximum Capacity Path* between two vertices in a graph

```
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
  unoptimized 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 neighbor 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[ ]
```

Algorithm 2 Greedy heuristic for computation of max-capacity routes between two vertices in a graph

```
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
```

fair distribution of the available energy resources among different source-destination pairs.

The algorithm has the following advantages: 1) It is distributed, enabling each sending-receiving actor node pair to make decisions independent of other sender-receiver pairs. 2) It achieves energy efficiency close to optimal value. 3) It provides fairness across multiple flows.

D. Modeling 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 (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, $f_i \leq f_i^*$.

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

$$\chi = \left(\sum x_i\right)^2 / \left(k * (\sum x_i^2)\right) \qquad (7)$$

where $x_i = f_i / f_i^*$ and $0 \leq i \leq k$.

The value of χ varies from $1/k$ (completely unfair) to 1 (completely fair).

IV. Energy Efficient Directional Routing

In this section, we discuss the design details of the EEDR protocol. The protocol achieves AAC in partitioned networks by utilizing 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 [3] and Ad Hoc On-demand Distance-Vector (AODV) routing [4] protocols.

A. Intra-Partition Routing

Within each partition, one actor node is always designated as the *leader* node. The leader election protocol, needs to be robust in the events of node mobility. In our protocol, we use the leader election algorithm described in [8]. The leader node is responsible for coordinating the actions of the actor nodes of that partition in the route discovery process. It enables the whole partition to act as a single entity to implement the algorithm presented in the previous section.

B. Inter-Partition Routing

When a source actor node needs to send data to a destination actor node, it looks into 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.

1) Route Discovery: The actor node first sends the route-request 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 route request packet to all actor nodes of the partition and initiates a directional broadcast.

2) 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 neighboring 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 neighbor. We assume that the actor nodes have one of the following capabilities: 1) Location awareness, or 2) Detecting the angle of arrival of a packet. The actor nodes perform this directional broadcast in the sensor-channel using their long-range directional antennas. Figure 2 illustrates the phenomenon of directional broadcast on a simple WSAN. The local decisions at each actor node within a given partition results in non-contending globally optimum broadcasts. As we can observe, the broadcasts happen in an outward fashion with minimal inter-sector overlap, making sure that there is no unnecessary wastage of the sensor node energy.

3) Sensor-to-Actor Routing: 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.

C. Route Selection

Once the destination actor node receives the packet, it sets off 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 Algorithms 1 and 2. It then informs the source node of these routes by reinforcing these routes using route-reply packets. Once, the source node receives a route-reply packet, it adds the route to its route cache.

Figure 3 shows the multi-path routing from the source node after destination node performs route selection. Due 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 the greedy heuristic described in section III-C.

V. Performance Evaluation

We evaluated the performance of EEDR using the ns-2 [9] network simulator.

A. Simulation Parameters and Metrics

Table I summarizes the various parameters and their settings in our simulation. The simulation setup consists of sensor nodes and actor nodes (divided into 6 partitions) randomly distributed in the field. 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 a node from each of these partitions for each run. The results presented are averaged over 10 runs for each set of parameters.

TABLE I: Simulation Parameters

Simulation area	1000 m x 1000 m
Number of sensor nodes	1000
Number of actor nodes	50
Number of dir. antenna sectors	4, 6
Number of actor flows	1, 2, 4
Sensor node transmission range	30 m
Actor node transmission range	150 m
Packet size	512 Bytes
Actor traffic pkt inter-arrival time	0.1 to 1 s
Actor speed	2 m/s
Mean event inter-arrival time	10, 20, 30, 40, 50 s

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 network lifetime as the time between the beginning of the functioning of the WSAN to the point at which there exist at least two actor nodes that can not communicate with each other.
- *Throughput* - We calculate the average throughput that each flow is able to achieve during the simulation.
- *Fairness* - Fairness has been defined in Section III-D. A fair protocol would lead to a fairness close to 1.0, whereas an unfair one would lead to a fairness close to $1/k$, where k is the number of flows.

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 4-sector and 6-sector setup respectively. In addition, we compare the performance of EEDR with that of flat AODV protocol [4]. In simulating AODV, we consider the actor and sensor nodes to be of equal capability and use short-range communication over a common channel. This would help in understanding the performance gain achieved by our proposed architecture and the efficiency of EEDR.

B. Static Topology

1) Network lifetime: The network lifetime of EEDR is compared with that of AODV and the results are presented in Figure 4. EEDR achieves as much as 80% improvement in lifetime when compared to AODV irrespective of number of simultaneous flows in the network. Also, the use of 6-sector directional antenna helps in improving the network lifetime to about 10% more than that of a 4-sector antenna.

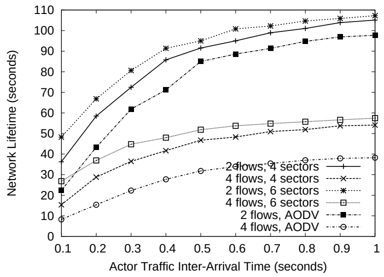

Fig. 4: Network lifetime (static topology) comparison of 6 and 4 sector EEDR with AODV

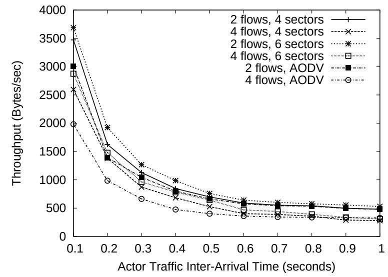

Fig. 5: Throughput (static topology) comparison of 6 and 4 sector EEDR with AODV

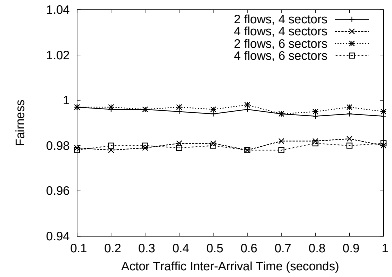

Fig. 6: Fairness (static topology) comparison of 6 and 4 sector EEDR

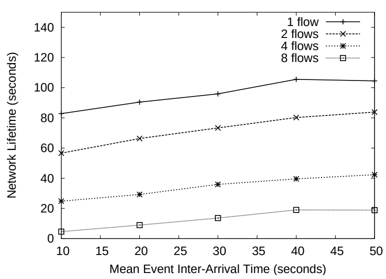

Fig. 7: Network lifetime (dynamic topology)

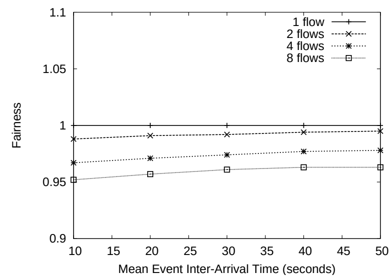

Fig. 8: Fairness (dynamic topology)

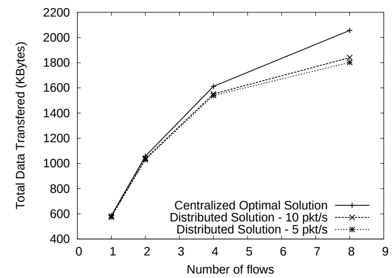

Fig. 9: Performance comparison of the distributed and centralized solutions

2) Throughput: In Figure 5, the results of the network throughput by EEDR and AODV are plotted by varying number of sectors and number of flows. The decrease in throughput with increase in packet inter arrival time is due to the packet drops that could occur at the sensor nodes. The results reveal that the EEDR with narrow beam-width achieves higher throughput.

3) Fairness: Figure 6 compares the cross-flow fairness obtained with actor antennas of two different sector angles - 90° (4-sector) and 60° (6-sector). The sector angle has little influence on the value of achieved fairness which remains high (close to 1.0) in both the cases.

C. Dynamic Topology

In this set of experiments, we simulate a dynamic topology caused due to actor node mobility and measure the performance of EEDR. We assume that the node movement is caused due to 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. Thus, a higher event-rate leads to more frequent actor node movements leading to a more dynamic topology.

1) Network lifetime: Figure 7 shows the variation of the network lifetime with mean inter-event arrival time for a network with various number of simultaneous flows. We observe that the network lifetime achieved is close to that in static topology and decreases with increasing number of flows due to increasing contention for the energy of the underlying network of sensor nodes. Also, for a given number of flows, the network lifetime increases with decreasing event-rate due to more frequent path breaks leading to more aggressive route discovery broadcasts.

2) Fairness: Figure 8 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 even under dynamic network conditions. The fairness decreases slightly with increasing number of flows and remains almost constant for varying event-rate.

D. Performance of the Distributed Solution

Figure 9 shows the comparison of the performance of EEDR with that of the centralized solution (Section III-B). The centralized solution is obtained by solving the optimization problem on the network graph using the General Algebraic Modeling System (GAMS) [10]. The performance of the distributed solution was measured by running EEDR for two different packet rates - 10 and 5 packets/second. 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 centralized solution. It is also to be noted that the difference in the performance of the two solutions is also partly due to the overhead of periodic route discovery, which is ignored in the centralized solution.

VI. Related Work

Energy efficient routing towards multiple sinks in a WSN is an important problem. “Anycast” [11] considers the problem of multi-mobile sinks. To route their data, the sensor nodes construct anycast trees in which the leaf nodes are the sinks. The sink nodes forward the received packet to the intended destination using an out-band channel. However, such an assumption of sink connectivity 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” [12] considers the case of diverting the traffic generated by sensor nodes to the Physical Sink via a set of resource rich Virtual Sinks in case of congestion notification. In case of partition in the Virtual Sink backbone network, the virtual sinks use the intermediate sensors to bridge the gap by switching to short-range radio, thus leading to increased end-to-end latency.

The use of directional antenna at sink node is proposed in [13] and [14] in order to extend the lifetime of the sensor nodes in relay zone. However, these works do not address the problem of multiple sinks and network partitioning.

In [15], the problem of optimal assignment of actor nodes in WSANs is formulated as an optimization problem such that the energy required for the mobility of actors is minimized while maximizing the number of events visited. However, this architecture relies on the presence of static agents at every zone in order to have undisturbed connectivity among them. The work presented in this paper, deals with design of an efficient routing protocol that utilizes the intermediate sensor nodes in bridging the actor partitions.

VII. Conclusions and Future Work

Actor-actor communication is an important part of the functioning of WSANs and enables the actors to take coordinated action on a given event. We propose to use the underlying sensor nodes to heal the actor network partitions. In order to maximize the utilization of the limited energy available with the sensor nodes, we propose a new routing protocol for AAC along with directional antennas on the actor nodes.

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

Unlike in SAC, AAC demands a completely reliable transport protocol. Thus, we intend to extend the work towards designing a reliable and low energy transport protocol for our proposed communications architecture.

References

[1] I. Akyildiz and I. Kasimoglu, “Wireless Sensor and Actor Networks: Research Challenges,” *Ad Hoc Networks*, vol. 2, no. 4, pp. 351–367, 2004.

[2] K. Selvaradjou, B. Goutham, and C. S. R. Murthy, “Exploiting Resource-Rich Actors for Bridging Network Partitions in Wireless Sensor and Actor Networks,” in *Proceedings of International Conference on Distributed Computing and Networking (ICDCN)*, January 2008.

[3] D. B. Johnson and D. A. Maltz, “Dynamic Source Routing in Ad Hoc Wireless Networks,” in *Mobile Computing*, vol. 353, Kluwer Academic Publishers, 1996.

[4] C. E. Perkins and E. M. Royer, “The Ad Hoc On-demand Distance-Vector Protocol,” *Ad Hoc Networking*, pp. 173–219, 2001.

[5] Y.-J. Kuo and H. D. Mittelmann, “Interior Point Methods for Second-Order Cone Programming and OR Applications,” *Computational Optimization and Applications*, vol. 28, no. 3, pp. 255–285, 2004.

[6] T. Tsuchiya, “A Convergence Analysis of the Scaling-Invariant Primal-Dual Path-Following Algorithms for Second-Order Cone Programming,” *Optimization Methods and Software*, vol. 11/12, pp. 141–182, 1999.

[7] B. V. Cherkassky, A. V. Goldberg, and T. Radzik, “Shortest Paths Algorithms: Theory and Experimental Evaluation,” in *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 516–525, 1994.

[8] A. Boukerche and K. Abrougui, “An Efficient Leader Election Protocol for Mobile Networks,” in *Proceedings of ACM International Conference on Wireless Communications and Mobile Computing (IWCMC)*, pp. 1129–1134, 2006.

[9] “The Network Simulator NS-2.” Available: http://www.isi.edu/nsnam/ns/.

[10] A. Brook, D. Kendrick, and A. Meeraus, “GAMS, a User’s Guide,” *SIGNUM Newsletter*, vol. 23, no. 3-4, pp. 10–11, 1988.

[11] W. Hu, S. Jha, and N. Bulusu, “A Communication Paradigm for Hybrid Sensor/Actuator Networks,” in *Proceedings of the 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 47–59, September 2004.

[12] C. Y. Wan, S. B. Eisenman, A. T. Campbell, and J. Crowcroft, “Siphon: Overload Traffic Management using Multi-Radio Virtual Sinks,” in *Proceedings of the 3rd ACM Conference on Embedded Networked Sensor Systems*, pp. 116–129, November 2005.

[13] J. Cho, J. Lee, T. Kwon, and Y. Choi, “Directional Antenna at Sink (DAaS) to Prolong Network Lifetime in Wireless Sensor Networks,” in *Proceedings of the 12th European Wireless Conference*, pp. 1–5, April 2006.

[14] S. Mao and Y. T. Hou, “BeamStar: A New Low-cost Data Routing Protocol for Wireless Sensor Networks,” in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM)*, vol. 5, pp. 2919–2924, 2004.

[15] K. Selvaradjou and C. S. R. Murthy, “On Maximizing the Residual Energy of Actors in Wireless Sensor and Actor Networks,” in *Proceedings of 8th International Conference on Distributed Computing and Networking (ICDCN)*, pp. 227–238, 2006.