SOFROP: Self-Organizing and Fair Routing Protocol for Wireless Networks with Mobile Sensors and Stationary Actors

Mustafa İlhan Akbaş*, Matthias R. Brust†, and Damla Turgut*
*School of Electrical Engineering and Computer Science
University of Central Florida, Orlando FL
Email: {miakbas,turgut}@eecs.ucf.edu
†Computer Science Division
Technological Institute of Aeronautics, Brazil
Email: matthias.brust@ita.br

***Abstract*—Wireless sensor and actor networks (WSAN) have become increasingly popular in recent years. The cooperative operation between sensor and actor nodes results in a major advantage against pure sensor networks and extends the range of possible application scenarios. One emerging application is the Amazon scenario in which stationary actors are deployed at accessible points in a thick forest structure and sensor nodes are thrown in a river flowing through the forest to gather observations from unreachable areas of the forest. This unprecedented and unique setting exposes two important challenges: (a) the dynamics of the river forms a continuously varying topology of sensor nodes requiring a highly adaptive network organization and (b) the inherent features of sensor and actor nodes, combined with rapid changes in the link structure of the network require efficient bandwidth utilization and data transmission.**

In this paper, we address these challenges by introducing SOFROP, a novel self-organizing and fair routing protocol for WSANs. The extensive simulations that are carried out for evaluation point out two highlights of SOFROP. These are the lightweight and efficient routing that is optimized for fairness and the locally acting adaptive overlay network formation.

I. Introduction

Wireless sensor and actor networks (WSANs) [1] consist of a large number of tiny sensor nodes, which observe the events in the environment, and limited number of more powerful actors, which collect information from the sensor nodes, process this information and react to the event. In contrast, in most of the pure wireless sensor network (WSN) [2] applications, the network is used only for observation, which is insufficient in many real-world scenarios. For instance, if sensor nodes are able to inform a patrol unit about their battery conditions, this unit should not only collect data but also move closer to the sensor nodes in need of recharging. Thus, WSANs provide functionalities that allow dealing with a wider range of possible application scenarios compared to WSNs.

Coexistence of actor and sensor nodes creates a heterogeneous structure of node resources and communication channels. Hereby, a sensor node has limited data processing capability, transmission rate, energy, and memory. Actors on the other hand possess increased computation capabilities and wider communication ranges. Usually, actors are also equipped with long lasting batteries and larger memory compared to sensor nodes.

In WSANs node mobility is a natural element of many applications such as intelligent transportation, environment monitoring, animal control. There are various communication algorithms in the literature that have been developed for scenarios in which the actors are mobile while sensor nodes are stationary (see [3] [4]). Although the sensor nodes are homogeneous, the network structure they form is fundamentally determined by the applications, which requires scenario-optimized network organization and routing schemes.

In context of this paper, we consider the scenario of Amazon rain-forest with a river going through it (see Fig. 1). The actors are positioned at rare accessible parts of the area and the sensor nodes are thrown in the river to collect information from hard-to-access sides of the area while they float in the river. Equipped with appropriate measurement technologies, sensor nodes are able to gather various kinds of information. For instance Regan et al. [5] deployed such a multi-sensor system in the River Lee Co. Cork, Ireland to monitor water quality parameters such as pH, temperature, conductivity, turbidity and dissolved oxygen. Although nodes move basically in one direction, they suffer from various peculiarities of the scenario such as permanent velocity changes, sudden stops by obstacles. In summary, these circumstances raise the following challenges for the design of an efficient routing protocol: (a) the dynamics of sensor nodes form a continuously varying topology requiring a highly adaptive network organization and (b) rapid changes of the neighborhood and actor association demands an efficient and reliable transmission of data from sensor nodes to the actors.

The limitation regarding the capacities of sensor nodes make the consumption of communication, computation and memory resources a critical constraint for routing protocols in WSANs [6]. An additional constraint arises with the heterogeneous node structure. Therefore the solutions applied neither in traditional wired networks nor in WSNs are suitable for WSANs. In

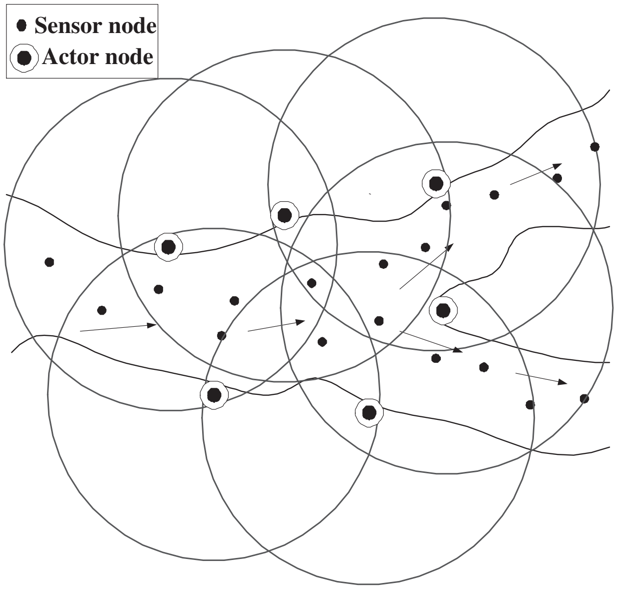

Fig. 1. Amazon application scenario.

our particular case, the Amazon scenario also comprise challenging and unprecedented characteristics such as the distinct mobility of the sensor nodes and constrained positioning of the actors. For these reasons existing approaches in the field of WSANs are inefficient and cannot be employed to overcome the variety of challenges implied by the differentiating features of the Amazon scenario.

The characteristics of Amazon scenario also create challenges for the quality of service provided to the applications in the network. Quality of service (QoS) in WSANs is characterized according to the employed applications, each of which has various constraints such as reliability, latency and robustness. QoS must be supported to satisfy the requirements of services of WSAN [7]. Throughput, delay, jitter, and packet loss rate are among the most fundamental QoS metrics used to measure the degree of efficiency in these services [8]. QoS performance of a network is improved in terms of these parameters when nodes allocate bandwidth in a fair manner [9]. For instance, even when there are multiple nodes reporting low priority traffic, which causes congestion on a bottleneck node, high priority data traffic must get its share of bandwidth and be transmitted to the relevant actor. All these requirements make fairness a vital part of the algorithm to be used. In the context of this paper, we define QoS as the capability to provide assurance that traffic flows will be treated differently in order to meet the service requirements of the applications.

In this paper, we propose SOFROP (Self-Organizing and Fair ROuting Protocol) to address the described critical issues of the Amazon scenario. There are two main contributions of this work. First, SOFROP uses an overlay network to organize and divide the network into actor areas. We show that SOFROP can provide a dynamic network organization by using locality-preserving communication. Second, SOFROP provides fairness among different types of applications while using the excess bandwidth at sensor nodes with a lightweight algorithm. Additionally, SOFROP uses packet queues to overcome the particular connectivity challenges of the scenario.

The remainder of this paper is organized as follows. Related work is given in Section 2. The system model is described in Section 3. We provide a detailed description of SOFROP in Section 4, which is devided in two phases: network organization and data transmission. We show the simulation results in Section 5 and finally conclude in Section 6.

II. Related Work

There has been approaches on routing protocols with QoS support in WSANs. The anycast communication paradigm by Hu et al. [10] builds an anycast tree rooted at each event source to reduce latency and energy consumption. The real-time routing framework by Shah et al [11] addresses the coordination of sensor and actor nodes through the delay bound for distributed routing. Another protocol using delay as the main constraint is Ad Hoc On Demand Delay Constrained Distance Vector Routing (AOD^{2}V) by Sama and Akkaya [12], where delay-EDD is used at admission control and EDF is used to determine the departure order of the packets at the intermediate nodes. Boukerche et al. [13] used service differentiation and central processing of routes, aiming low latency and reliable delivery. Paruchi et al. [14] proposed a distributed and randomized communication protocol with a fairness feature regarding power savings of the sensor nodes, which make local decisions on whether to sleep, or be active based on the energy levels of their neighbors. Xia et al. [15] applied feedback control for dynamic bandwidth allocation, which uses deadline miss ratio control to improve QoS in terms of reliability.

Another important asset of SOFROP is the employment of an overlay network scheme for the network organization part by clustering. Clustering algorithms in traditional sensor network, are often used to create a structure of an otherwise flat network topology [16], [17]. A cluster is a group of interconnected nodes with a dedicated node called clusterhead. A clusterhead is in charge to manage the information flow in the proper cluster, which makes the role and the selection of clusterheads critical for proper network operation. One of the clustering scheme proposed for sensor networks is HEED by Younis and Fahmy [18], in which authors target QoS by using an efficient energy consumption method.

Network organization behaviors of Amazon scenario requires the actors to be assigned as the pre-determined clusterheads, which are not supposed to change their status throughout the life-time of the network as it occurs in any other existing clustering algorithm for mobile ad hoc networks [19]. Thus the clusterhead election procedure is obsolete. However, the network and the clustering algorithm must be designed in a way that the actor node is always the most attractive clusterhead in its surrounding. Furthermore, the cluster structure permits multi-hop clusters, which is supported by only a few clustering algorithms. Since actors are specially equipped to process data while delivering a long life-time, the number of actors must be minimized. This property reduces the number of clusterheads in the network, which is also important for the case where actors cannot be deployed very close to each other due to restricted access to the environment. A characteristic of the Amazon scenario is that the actors are static while the sensor nodes are mobile. This causes an increased number of re-affiliations for the sensor nodes, since the clusterhead for a node is expected to change frequently. These distinct features of the application scenario require a novel network organization approach. Therefore a locally acting clustering approach, which adopts its basic principles from the multi-hop clustering scheme KHOPCA [20], is used and it is described in Section 4.

The approaches discussed in this section do not match with the requirements of the Amazon scenario such as simplicity, flexibility and fairness, caused by the dynamics of the application scenario and the characteristics of WSANs. The routing algorithm in this scenario should be able to adapt to the rapid changes in the topology and allocate the bandwidth in a fair manner for QoS support.

III. System Model

We consider a wireless sensor and actor network N with the number of nodes $|N| = n$. N consists of a set of actor nodes A and a set of sensor nodes S. Additionally our model includes a sink node in charge of data aggregation and connection to a backbone network. The sensor nodes and actors in S are assumed to have maximum transmission ranges r_s and r_a, respectively, with circular transmission areas, where $r_s < r_a$ due to better computation and communication capabilities of the actors. For communication between two nodes, a bidirectional connection must be established, i.e. a device $s1$ has to be in the transmission range of $s2$, i.e $d(s_1, s_2) \leq r_s$, for communication.

A. Sensor nodes

For each sensor node s in S let be a neighboring list $Neigh(s) \subset N$, the set of nodes that are directly connected to s, such that $\forall u \in Neigh(s), d(s,u) \leq r_s$. $Neigh(s)$ is built initally when a node enters the network and updated with a given update frequency f.

Every node is able to communicate only with its current one-hop neighbors (a sensor node or an actor), thus all communication in our model is locality preserving. Geographical positions of the nodes are not used. Since data flows in only one direction and only local information are used, no multi-hop control communication need to be applied. Communication links may fail or disappear from the network caused by obstacles for instance. Thus, the neighborhood of a node changes over time and nodes move with random and nonconstant speed, acceleration and directions.

B. Actors and the sink

SOFROP manages actor-actor communication efficiently in order to save battery lifetime. The actor nodes use their full transmission range in two cases only. One of these cases is the time when the network is initialized, in which the actor nodes and the sink create a network by using their full transmission ranges. In our particular application scenario, the actor nodes are positioned such that each one has at least one actor or sink in its transmission range. The sink communicates only with actors and it is also positioned in the transmission range of at least one actor. Otherwise the sink would be required to receive the collected data through the sensor nodes, which would create severe packet loss and delay conditions in the network. Considering this layout and the small number of actor nodes, the following steps are taken to form the links among actors and the sink:

- The sink starts the formation of links by flooding its ID and hop count (initialized as 1) encoded in a packet.
- This packet is forwarded in the network among actors and each actor saves the ID of the actor from which it received the packet with the lowest hop count as the destination for data traffic.
- The packet is retransmited with an incremented hop count only if its hop count is less than the actor's.

The other case when an actor uses its full transmission range is when it has data to exchange, consolidate and transmit to the sink. Other than these two cases, actors use the same transmission range as the sensor nodes in the network organization phase and in communication with the sensor nodes. Therefore this approach extends the lifetime of the actors, which is an energy-efficient feature of SOFROP. However, it is important to note that the actor-actor communication is not the main focus of this paper and will be tackled more in detail in future work.

IV. Self-Oriented and Fair Routing Protocol

The separation of the network organization from data transmission shows several benefits since the network organization phase adjusts the topology of the sensor nodes to enable efficient routing on the resulting overlay network. This separation reduces route failures and packet delay, while increasing the network throughput [21]. Hence, SOFROP is divided into two phases: the first phase concerns about the network organization, where an overlay network is formed and continuously adapted. The second phase is responsible for the data transmission.

A. Network organization

The clustering is used for the network organization. However, in contrast to common environments where clustering algorithms are applied, SOFROP has to deal with the fact that actor nodes are pre-assigned clusterheads. Additionally, due to restrictions in the deployment of actor nodes, multi-hop clusters must be created as a remedy. Furthermore, since actor nodes are specially equipped while delivering a long life-time, the number of actor nodes must be minimized. The mobility of sensor nodes increases the number of re-affiliations to the actors. An additional modification is the resulting overlay network providing a topology, allowing multiple paths to the actor-actor network.

1) SOFROP overlay network setting: The algorithm to create the overlay network does not require any initial configuration besides that each node must choose a value between 1 and k, its weight. The weight 0 is exclusively assigned to the actors. We assume that the only information available for a sensor node s is the information of the direct neighbors $Neigh(s)$ and their corresponding weights $w(Neigh(s_i))$. The beaconing is the most common way to provide this information. However if beacon (or heartbeat) approach is used in the network, then the sensor nodes are required to transmit a packet periodically even when there is no neighbor node to receive this packet. Although beaconing is commonly used in sensor networks, it should be avoided when possible since energy is one of the most important constraints of sensor nodes. In order to attend the energy requirements of the Amazon scenario, we propose a different approach to transfer weight information:

- Only the actor nodes generate packets periodically. These packets are called *Area Configuration Packets* (ACP), which includes actor ID and hop count fields. The actor initializes these fields with its ID and hop value respectively.
- A sensor node receiving the ACP drops the packet if the hop value on the packet is greater than or equal to its own hop value. Otherwise the node stores the values in actor ID and hop value fields of the packet and retransmits the packet with an incremented hop value.
- If a node looses connection to the actors, it equals its hop value to the maximum hop value defined for the network. A node loses connection when it doesn't receive an ACP (either directly from an actor or by retransmission of other nodes) for the predefined time in the network.

The nodes that lose the connection will be only in "listening" mode and they won't transmit any packets while actor nodes periodically send ACPs. This structure makes optimum use of the WSANs since the complexity and resource requirement is focused on the actor nodes and it is less energy consuming than beaconing for the sensor nodes.

2) SOFROP clustering algorithm: Consider a node v with weight $w(v)$. The state transition for node v is given in Algorithm 1:

Algorithm 1 The state transitioning of a node v

```
$min.weight$= $w(v)$
for $i \in Neigh(v)$ do
    if $w(i) < min.weight$ then
        $min.weight = w(i)$
    end if
end for
if $min.weight < w(v)$ then
    $w(v) = min.weight + 1$
else if $w(v)! = k$ then
    $w(v) = w(v) + 1$
end if
```

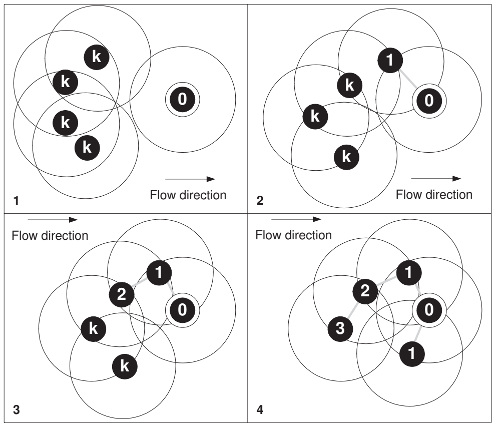

Fig. 2. A sequence example for 4 sensor nodes and an actor.

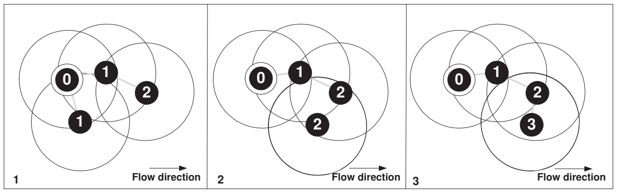

Fig. 3. A sequence example of a small group of sensor nodes and an actor

An example sequence for four sensor nodes and one actor node is demonstrated in Fig. 2. In the beginning of the sequence, the actor node has a weight of zero and all sensor nodes are initialized with the weight value k. The remaining sequence shows how the hierarchical structure is formed by copying the lowest neighbor weight added by one as the sensor nodes get into transmission range of the actor.

When a lower weight node that is not an actor attracts surrounding nodes with higher weights, this node successively increases its weight in order to avoid a fragmented structure. An example of this property is demonstrated in Fig. 3, where one of the node is moving faster compared to the other nodes in the scenario, which is a possible case due to potential obstacles and unpredictable flow rate changes in the river. The fast-moving node initially has a weight of "1" since it is directly connected to the actor at the beginning, but it loses its connection to the actor after it gets further away. However it still receives ACPs since it is in the transmission range of a node with weight "3". Then it increases its weight to connect to the closest actor.

A sequence of the dynamic overlay network produced by network organization is denoted in Fig. 4. Three actors are deployed uniformly at random and remain static while 60 sensor nodes are flowing from left to right, where the maximum hop-count allowed by the network organization is 4. Note that Figure 4 depicts only one of the different outcomes which are possible due to asynchrony.

Isolated nodes with weight k are physically able to communicate with neighbors having weight k, but according to

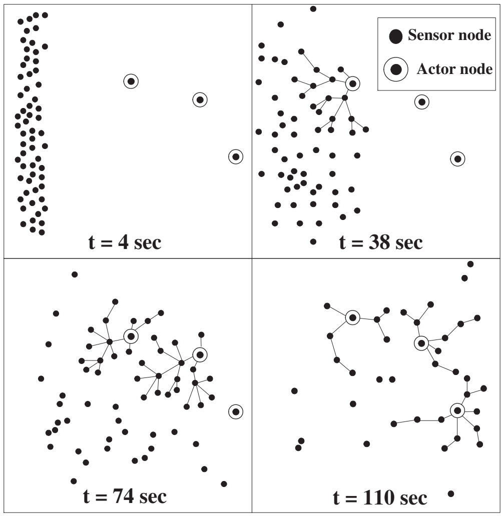

Fig. 4. A squence of the dynamic overlay network produced by SOFROP network organization.

our network organization phase no logical communication path is built. In order to include the isolated nodes in the network, a remedy can be to increase the difference of 0 and k, forcing this phase to built longer communication paths. The network designer, however, has to consider the velocity and perturbations of the river that act on the nodes. If the paths become extremely long, no effective routing can be conducted or the messages from the most distant node may never reach the actor node. For that reason, the difference between 0 and k must be chosen according to the environmental conditions. Thus, sensor nodes outside the coverage area of the actor nodes are simply ignored and they do not influence the remaining network due to their k-weight.

This phase uses only local information for decision making process and all the nodes rapidly updates their data as the network structure changes.

B. Data transmission

The information collected from the environment and aggregated in the network is defined as the "interests" of the sink. This term is adopted from the language popularized by the Directed Diffusion model [22]. The interests are predefined at sensor nodes before they are thrown into the river. This method is efficient considering the dynamic topology of the network and hard-to-estimate mobility patterns of the sensor nodes. Each sensor contains a list of the interests, called the "interest table", which is used when an event is captured to decide whether the sink needs to be notified. The interest in the table for a captured event is called an *on* interest and the node generates data packets to report the event to the closest actor. The data is transferred in fixed size packets and we define the time required to transmit one packet as one "time window". The maximum packet transmission rate of a sensor node is called output capacity (C_o), which is reduced by the rate value on the first packet of an interest and this value is recorded as the remaining output capacity (C_r) of the node.

The SOFROP's main goal for data transmission is the fair allocation of network resources among flows of different interests. This is achieved without keeping any global state at sensor nodes. Instead, a node capturing an event encodes data packets with the rate it transmits them (α_p) and the interest (i_p) that the packets belong to.

Sensor node buffers data in first in first out (FIFO) fashion, which supports the idea of keeping sensor nodes as simple as possible. There are n input links of a sensor node for receiving data and one output link for forwarding data. We assume that the packet arival process at each input link is a Bernoulli process with success probability p_s, the number of packet arrivals (A) at the buffer during a given time window has the binomial probability mass function and the probability generating function of the Bernoulli random variable with parameter p_s is as follows:

$$G_A(z) = \sum_{i=0}^{n} a_i z^i = (1 - p_s + p_s z)^n$$

A sensor node does not drop any packets when $C_r \geq 0$, which means the sensor node's resources are adequate to serve the received packets. When $C_r \geq 0$, the number of packets in the buffer at the end of the k^{th} time window (B_k) can be defined in terms of the number of packets in the buffer at the end of the $(k-1)^{th}$ time window and the number of packets arriving during the k^{th} time window (A_k) as $B_k = max(0, B_{k-1} + A_k - 1)$. The underlying stochastic process of B_k can be described by a Discrete Time Markov Chain (DTMC) with states $q_i = P(N = i)$ [23]. If the sensor node does not drop any packets for a period of time, then it means $np_s \leq 1$ in one time window of this period. Consequently, the buffer occupancy can be formulated as $B_k = max(0, B + A - 1)$. Then its probability generating function (pgf) is found as:

$$\begin{aligned} G_B(z) &= \sum_{j=0}^{\infty} P(B = j) z^j \\ &= \sum_{j=0}^{\infty} q_j z^j \\ &= a_0 q_0 + \sum_{j=0}^{\infty} P(B + A - 1 = j) z^j \\ &= a_0 q_0 + \frac{G_B(z) G_A(z) - a_0 q_0}{z} \\ &= \frac{a_0 q_0 (z - 1)}{z - G_A(z)} \end{aligned}$$

The probability generating function satisfies $G_B(1) = 1$. We also have $\lim_{z \to 1} a_0 q_0 (z-1) = \lim_{z \to 1} z - G_A(z) = 0$. Thus we can apply l'Hopital's rule and get:

$$1 = G_B(z) = \frac{a_0 q_0}{1 - G'_A(1)} = \frac{a_0 q_0}{1 - np_s}$$

Therefore $a_0 q_0 = 1 - np_s$. Then:

$$\begin{aligned} G_B(z) &= \frac{(1-np_s)(1-z)}{G_A(z)-z} \\ &= \frac{(1-np_s)(1-z)}{(1-p_s+p_s z)^n - z} \end{aligned}$$

The expected value of $G_B(z)$ is equal to the mean steady-state queue size of the buffer, which is found by differentiating $G_B(z)$ with respect to z and taking the limit as $z \to 1$:

$$E(B) = \frac{n(n-1)p_s^2}{2(1-np_s)} = \frac{(n-1)}{n}\frac{(np_s)^2}{2(1-np_s)}$$

We define a fair rate (α_f) value, which is the amount of output capacity that the node can fairly employ for a flow when C_r is negative. The fair rate for a node depends on the number of on interests (N_i) and defined as $\alpha_f = C_o/N_i$.

However with this tagging, when C_r is negative there are flows with rate values lower than α_f. In such a case, if all packets are encoded with rate values smaller than or equal to α_f, an unutilized excess capacity is formed. The output capacity shared among flows that are received with rates greater than α_f is defined as the shared capacity (C_s).

When C_r is negative but the rate of the packet is smaller than α_f, the packet is forwarded without changing the values in its fields. If the rate tag on a data packet is greater than the fair rate, then it means the packets of the interest are received with a rate greater than the node can transmit, which will result in packet drops. In order to insert an exact rate value in the packets, number of transmitted and dropped packets must be recorded at the sensor node for a period of time. Limited memory and computation resources of a sensor node would be insufficient to keep such a state information for each flow. Therefore SOFROP drops packets probabilistically at each node depending only on the tags. The probability to drop a packet (P_d) is defined as follows:

$$P_d = 1 - C_s/(N_s \cdot \alpha_p)$$

where N_s is the number of the interests that shares C_s.

If a packet is not dropped when C_r is negative, then it is forwarded with a new α_p. The interest of this packet is defined as a sharing interest and this interest's share from C_s is the new α_p, which is defined as follows:

$$\alpha_p = \frac{C_s \cdot \rho_i}{\sum_{j=0}^{N_s} \rho_j}$$

The SOFROP allows assignment of different priorities to the interests whereby ρ_i is the priority of the current interest. The pseudocode of the algorithm used at sensor nodes for routing packets is given in Algorithm 2.

V. Simulation Study

A. Simulation environment and metrics

The simulation is conducted in the OPNET modeler [24]. IEEE 802.11 is the underlying MAC layer for sensor and actor nodes. The transmission range of a sensor node is taken as 40 meters, a realistic range for a sensor node (for instance Cerpa et al. [25] finds the transmission range of second generation Mica-2 motes to be between 20 and 50 meters in an outdoor habitat). The assumptions include a 20 packet-size queue and a data rate of 10 packets per second. Wireless LAN model in OPNET allows transmission power of a node to be defined as an attribute by means of OPNET's transceiver pipeline implementation. The relation between the transmission power of a node (T in Watts) and its transmission range (r) is defined as $T = \left(\frac{4\pi r}{0.12476}\right)^2 \cdot 10^{-12.5}$. Table I summarizes the simulation parameters used in our experimental setup.

Algorithm 2 Routing in a sensor node

```
if a packet is received by a sensor node then
    if the packet is a notification to a predefined interest then
        if it is the first packet for that interest then
            sharingInterest ← true for the interest
            increment $N_f$ by one, reduce $C_r$ by $\alpha_p$
        end if
        if $C_r > 0$ then
            forward the packet
        else
            if $\alpha_p > \alpha_f$ then
                toDropInterest ← true for the interest
                if sharingInterest: false then
                    increase $C_s$ by $\alpha_p$
                    sharingInterest ← true for the interest
                end if
                drop the packet with its $P_d$
                if the packet is not dropped then
                    fill the rate field and forward the packet
                end if
            else
                forward the packet
                if sharingInterest: true then
                    reduce $C_s$ by $\alpha_p$
                end if
                toDropInterest ← true, for the interest
                sharingInterest ← false for the interest
            end if
        end if
    else
        drop the packet
    end if
end if
```

TABLE I
Simulation parameters

Number of sensor/actor nodes	60/4
Total area	200x300
Reception power threshold (dBm)	-95
Channel settings	Auto assigned

The communication graph is built according to the system model specified in Section III. The communication links that

TABLE II
MOBILITY SETTINGS

Starting point	x= 0-10 m; y = 0-300 m
Destination point	x= 100-200 m; y = 0-300 m
Pausing point, time	Random, 0-10 sec
Speed	0-3 m/sec

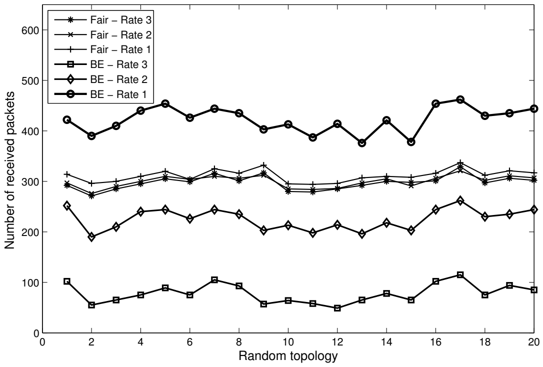

Fig. 5. Number of packets received by actors

sensor nodes form may fail or disappear from the network. A random mobility profile is created in OPNET modeler for the sensor nodes so that the nodes are moving in the watercourse with the settings given in Table II.

According to the optimization criteria, the efficiency of the proposed algorithm is studied using the following metrics: fairness, number of received packets, maximum hop value in the network and the number of nodes.

B. Simulation results

1) Experiment 1: 20 traffic sources are placed to generate three different types of packets (50% type-1, 30% type-2, 20% type-3) with equal priorities, which create congestion and bottlenecks in the network depending on the dynamic topology. The same set of simulations is run with and without the fairness property of SOFROP. The number of received packets for each type of traffic is denoted in Fig. 5. Packet drops occur due to high packet generation rate. SOFROP with fairness drops more packets of the type with larger rate, which is favorable in the Amazon scenario since one traffic type should not suppress the others. However the same property cannot be observed in best effort case, i.e. without fairness. Additionally, the total number of received packets is smaller.

2) Experiment 2: The same simulation set is run with slight changes, where 50% of the produced packets are type-1, 45% are type-2 and the rest is type-3. The priority of type-3 is three times larger than the other types, which shows the criticality of type-3 information. The number of packets received by the actors for each type is denoted in Fig. 6. Results show that SOFROP protects the critical type of traffic even when there is bursty non-critical traffic.

3) Experiment 3: The delay characteristics of SOFROP are observed by using a similar simulation set in experiment 2. Delay values depend fairly on topology in our application scenario since the path of a packet changes with the topology and the number of actors. It is shown in Fig. 7 that SOFROP performs clearly better when it is fair, which is critical when combined with the previous results. The results indicate that SOFROP not only protects critical packets but also delivers packets with a low average delay, which is another main QoS parameter.

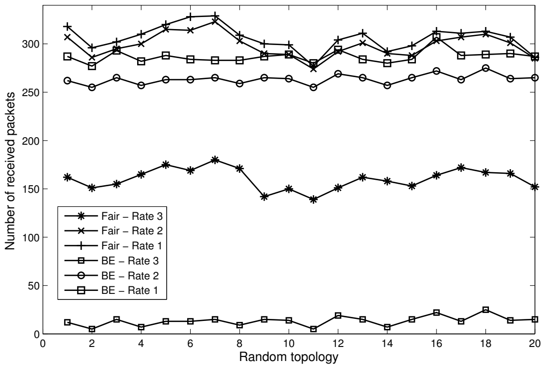

Fig. 6. Number of packets received by actors

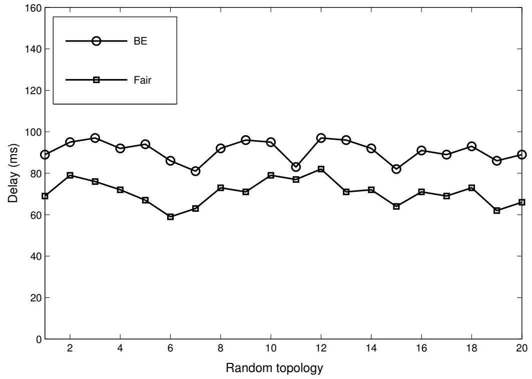

Fig. 7. End-to-end delay

4) Experiment 4: The SOFROP's coverage properties are investigated using the same simulation settings as in the previous experiments and the number of connected and unconnected nodes is observed in this experiment. Besides we measured the hop distribution for k values in between 3 and 6. For each value of k, 25 simulations are run and each simulation period ends as the first sensor node moves out of the area. The average numbers of sensor nodes with different hop-count values are presented in Fig. 8 for each k value. The results show that number of unconnected nodes decreases by 20 to 30% as k is incremented by 1. The number of nodes associated with an actor increases with increasing k; for example the average number of unconnected nodes is 20 when $k = 5$. Fig. 8 also shows that at least 45% of the nodes are in 2-hops distance for all values of k. Along with the other simulations, this experiment also denotes high adaptability of SOFROP's network organization to mobility.

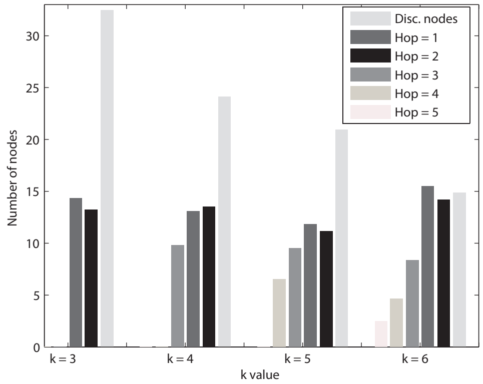

Fig. 8. Number of sensor nodes and their hop values for different k values

VI. Conclusion

In this paper, we propose SOFROP, a self-organized routing protocol that provides QoS for wireless sensor and actor networks. In particular, we focus on the Amazon scenario where actor nodes are deployed on few accesible locations and sensor nodes are deployed in the river to make measurements. The main characteristic of this scenario is the fact that the actor nodes remain static but irregularly deployed while the sensor nodes are moving in an unpredictable pattern according to the dynamics of the river.

The SOFROP is designed for this unique environment and consists of two phases: 1) the network organization builds a structured network topology that permanently adapts according to the river dynamics, and 2) in data transmission phase, data is collected from the network and forwarded to the sink. We show that SOFROP provides fairness among different types of applications and it is efficient in utilization of the bandwidth. Additionally, an actor and only 1-hop local information are provided to the sensor nodes. These features make SOFROP an ideal routing protocol for wireless networks with mobile sensors and stationary actors. As future work, we plan to adapt SOFROP to animal monitoring and conduct field tests in real life settings. We also intend to improve the network structure by permitting additional communication for sensor nodes along with data aggregation in order to increase redundancy and adaptability in the network.

VII. Acknowledgement

The authors would like to thank OPNET Technologies Inc. for supporting this research by providing OPNET Modeler software under OPNET University Program.

References

[1] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: research challenges," *Ad Hoc Networks*, vol. 2, no. 4, pp. 351–367, October 2004.

[2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[3] Y. Gu, D. Bozdağ, R. W. Brewer, and E. Ekici, "Data harvesting with mobile elements in wireless sensor networks," *Computer Networks*, vol. 50, no. 17, pp. 3449–3465, December 2006.

[4] A. Boukerche, M. Ahmad, B. Turgut, and D. Turgut, "A taxonomy of routing protocols in sensor networks," in *Algorithms and Protocols for Wireless Sensor Networks*, A. Boukerche, Ed. Wiley, 2008, ch. 6, pp. 129–160.

[5] F. Regan, A. Lawlor, B. Flynn, J. Torres, R. Martinez-Catala, C. O'Mathuna, and J. Wallace, "A demonstration of wireless sensing for long term monitoring of water quality," in *Proc. of the IEEE Conference on Local Computer Networks (LCN)*, December 2009, pp. 819–825.

[6] D. Peleg, *Distributed computing: A Locality-Sensitive Approach*. Society for Industrial and Applied Mathematics, 2000.

[7] F. Xia, "QoS challenges and opportunities in wireless sensor/actuator networks," *Sensors*, vol. 8, no. 2, pp. 1099–1110, February 2008.

[8] Y. Li, C. S. Chen, Y.-Q. Song, and Z. Wang, "Real-time QoS support in wireless sensor networks: a survey," in *Proc. of the 7th IFAC International Conference on Fieldbuses & Networks in Industrial & Embedded Systems - FeT'2007*, November 2007.

[9] J. Nagle, "On packet switches with infinite storage," *IEEE Transactions On Communications*, vol. 36, pp. 435–438, April 1987.

[10] W. Hu, N. Bulusu, and S. Jha, "A communication paradigm for hybrid sensor/actuator networks," in *Proc. of the 15th IEEE Int. Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, September 2004, pp. 201–205.

[11] G. A. Shah, M. Bozyigit, O. B. Akan, and B. Baykal, "Real-time coordination and routing in wireless sensor and actor networks," in *Proc. of Next Generation Teletraffic and Wired/Wireless Advanced International Conference (NEW2AN)*, May 2006, pp. 365–383.

[12] A. Sama and K. Akkaya, "Real-time routing for mobile sensor/actor networks," in *Proc. of the IEEE Conference on Local Computer Networks (LCN)*, October 2008, pp. 821–828.

[13] A. Boukerche, R. B. Araujo, and L. Villas, "A novel QoS based routing protocol for wireless actor and sensor networks," in *Proc. of IEEE Global Telecommunications Conference (GLOBECOM)*, November 2007, pp. 4931–4935.

[14] V. Paruchuri, A. Durresi, and L. Barolli, "Energy aware routing protocol for heterogeneouswireless sensor networks," in *Proc. of Database and Expert Systems Applications*, August 2005, pp. 133–137.

[15] F. Xia, Y. Sun, Y.-C. Tian, M. O. Tadé, and J. Dong, "Fuzzy feedback scheduling of resource-constrained embedded control systems," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 2, pp. 311–321, May 2009.

[16] K. Dasgupta, K. Kalpakis, and P. Namjoshi, "An efficient clustering-based heuristic for data gathering and aggregation in sensor networks," in *Proc. of Wireless Communications and Networking Conference, WCNC 2003*, vol. 3, March 2003, pp. 1948–1953.

[17] H. Luo, F. Ye, J. Cheng, S. Lu, and L. Zhang, "Ttdd: two-tier data dissemination in large-scale wireless sensor networks," *Wireless Networks*, vol. 11, no. 1-2, pp. 161–175, January 2005.

[18] O. Younis and S. Fahmy, "Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 1, March 2004, pp. 629–640.

[19] S. Basagni and E. Jonsson, "Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks," in *In Proc. of the IEEE 50th International Vehicular Technology Conference, VTC*, 1999, pp. 889–893.

[20] M. R. Brust, H. Frey, and S. Rothkugel, "Adaptive multi-hop clustering in mobile networks," in *Mobility '07: Proc. of the 4th International Conference on Mobile Technology, Applications, and Systems and the 1st International Symposium on Computer Human Interaction in Mobile Technology*, September 2007, pp. 132–138.

[21] N. Burri, P. von Rickenbach, R. Wattenhofer, and Y. Weber, "Topology control made practical: Increasing the performance of source routing," vol. 4325, no. 2, pp. 1–12, December 2006.

[22] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *MobiCom'00: Proc. of the 6th Annual International Conference on Mobile Computing and Networking*, August 2000, pp. 56–67.

[23] K. S. Trivedi, *Probability and statistics with reliability, queuing and computer science applications*. Chichester, UK: John Wiley and Sons Ltd., 2001.

[24] "Opnet modeler," http://www.opnet.com.

[25] A. Cerpa, N. Busek, and D. Estrin, "Scale: A tool for simple connectivity assessment in lossy environments," *Technical Report, Center for Embedded Network Sensing (CENS)UCLA*, June 2003.