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

Mustafa İ. Akbaş*, Matthias R. Brust†, and Damla Turgut*

*Department of EECS, University of Central Florida

4000 Central Florida Avenue, Orlando FL 32816

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 combined operation between sensor nodes and actors results in a major advantage compared to pure sensor networks extending the range of possible applications. One of the emerging applications 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. 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 requiring efficient bandwidth utilization and data transmission.

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

Keywords: Wireless Sensor and Actor Networks, Quality of Service, Fairness, Clustering.

I. INTRODUCTION

Wireless sensor and actor networks (WSANs) consist of a large number of tiny sensors and a limited number of more powerful actor nodes. The sensor nodes observe the events in the environment, and the actors are responsible to collect information from the sensors, process and react to an event [1]. In contrast, pure wireless sensor network (WSN) [2] applications are limited to observation only. Thus, WSANs deal with a wider range of possible application scenarios compared to WSNs.

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

WSNs and WSANs are employed for applications such as intelligent transportation, environment monitoring and animal control. The vision of a traditional WSN assumed both the sinks and the sensor nodes to be static [3], [4]. However node mobility is a natural element of many applications in WSANs [5], [6]. In the literature there are various communication algorithms developed for scenarios in which the actors are mobile while sensor nodes are stationary [7], [8]. Although the sensor nodes are homogeneous, the complex network they formed is fundamentally determined by the application. This is a fact that requires scenario-optimized network organization, data aggregation and routing schemes.

In the 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, while the sensor nodes are thrown in the river. Equipped with appropriate measurement technologies, sensor nodes are able to gather various kinds of data while floating in the river. Regan et al. [9] 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, etc.

In summary, these circumstances rise the following challenges for the design of an efficient routing protocol: (a) rapid changes of the neighborhood and actor association demands an efficient and reliable transmission of data from sensor nodes to the actors and (b) the dynamics of sensor nodes form a continuously varying topology requiring a highly adaptive network organization.

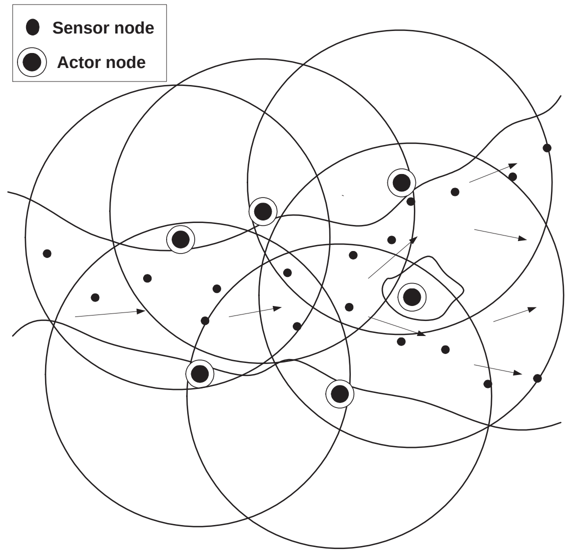

Fig. 1. Amazon River application scenario.

The sensor nodes have limited capabilities in terms of communication, computation and memory resources. Therefore the consumption of these resources is a critical constraint for routing protocols in WSANs [10]. This requirement makes QoS and fairness vital parts of the chosen algorithm. An additional constraint for routing protocols in WSANs arises with the coexistence of heterogeneous node resources. Therefore solutions applied in networks with rich resources or in WSNs are inconvenient for WSANs. In our particular case, the described setting for the Amazon scenario also comprises challenging and unprecedented characteristics such as the distinct mobility of the sensor nodes and constrained positioning of the actors.

In this paper we propose SOFROP (Self-Organizing and Fair Routing Protocol) to address in particular the following critical issues of the Amazon scenario:

Timely and fair data transmission: The QoS in WSANs is characterized according to the employed applications, each of which has various constraints such as reliability, latency and robustness. QoS performance of a network is improved in terms of these parameters when nodes allocate bandwidth in a fair manner [11]. For instance, even when there are multiple nodes reporting low priority traffic, which causes congestion on a bottleneck node, high priority data traffic must receive its share of bandwidth to be transmitted to the relevant actor. 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.

Indeterministic dynamics: When deployed in the river, the sensor nodes are subject to mobility. The sudden changes in current speed and direction combined with potential obstacles makes connectivity crucial among the nodes and the mobility pattern predictable only to a certain degree. Furthermore, the actor nodes can only be positioned on land and the sparsely accessible environment often impedes deploying actor nodes according to an ideal model, which would guarantee full connectivity at all times.

Restricted device deployment: Due to the node deployment restrictions the network structure has to allow multi-hop communication, i.e., sensors that are not directly connected to an actor should be allowed to communicate with other nodes to reach the actors. However, due to the dynamics of the river, routing paths continuously change and network re-organization occurs frequently. Therefore, the network organization should be done locally, avoiding superfluous message exchange, and it also must enable efficient realization of the routing protocol.

Utilization of resources: As a result of the scenario, the efficient utilization of the available bandwidth and minimization of packet loss become critical. Each sensor node must use its maximum packet transmission rate to forward as much information as possible. Therefore management of packet drop mechanisms and packet queues are two essential parts of the algorithm.

There are *two main contributions* of the proposed algorithm. 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.

The remainder of this paper is organized as follows. Related work is given in Section II. In Section III, the system model is explained and a detailed description of SOFROP is provided. We show the simulation results in Section IV and finally conclude in Section V.

II. Related Work

WSANs are deployed in different infrastructures with diverse applications. These different applications must be handled profoundly for an efficient operation of the network. Therefore Quality of Service (QoS) becomes a critical part of the communication protocols used in WSANs [12], [13]. Throughput, delay, jitter, and packet loss are among the most fundamental QoS metrics used to measure the degree of efficiency in these services [14], [15], [16].

There have been efforts on routing protocols that provide QoS support in WSANs. The anycast communication paradigm by Hu et al. [17] builds and updates an anycast tree rooted at each event source to reduce end-to-end latency and energy consumption. The real-time routing framework by Shah et al. [18] addresses the coordination of sensor and actor nodes and uses the delay bound for routing. The sinks dynamically join or leave and these actions affect the structure of the anycast trees. 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 [19], where delay-EDD is used at admission control and EDF is used to determine the departure order of the packets at the intermediate nodes. Morita et al. [20] proposed a redundant data transmission protocol, in which a sensor node sends its messages with sensed data received from other nodes additional to its sensed data to enhance the reliability of data transmission. Boukerche et al. [21] used service differentiation and central processing of routes, aiming at low latency and reliable delivery in the presence of failures. In this algorithm, the route generation is done at actors by using the information collected from the sensor nodes. Boukerche et al. [22] also used this approach additional to an energy-aware event-ordering algorithm to find a solution for context interpretation through a WSAN, in which actors are used to aggregate time correlated events from the sensor nodes and eliminate ambiguities. The distributed and randomized communication protocol by Paruchi et al. [23] has a fairness feature regarding power savings of the sensor nodes where they make local decisions on whether to sleep or be active based on the energy level of their neighbors. Xia et al. [24] applied feedback control for dynamic bandwidth allocation, which uses deadline miss ratio control to improve QoS in terms of reliability. QoSNET by Houngbadji and Pierre [25] takes the network lifetime as the main metric and formulates the QoS routing in large scale wireless networks as an optimization problem to extend the network lifetime. The management protocol for reactive sensor and actor systems by Baunach [26] focuses on memory and offers a collaborative approach.

These existing approaches take different requirements into consideration from those taken by the Amazon scenario. The dynamics of the application scenario of SOFROP when combined with the natural characteristics of WSANs requires a simple yet effective QoS support at the sensor nodes. The routing protocol suitable for Amazon scenario must adapt to the rapid changes of the network topology and allocate bandwidth according to the priority and the rate of the traffic.

Another important asset of SOFROP is the employment of an overlay network scheme for the network organization. Since the actor nodes are dedicated nodes collecting data from sensor nodes, a clustering approach is suitable for the network organization part. The clustering algorithms in traditional sensor networks [27] are often used to create a structure of an otherwise flat network topology [28], [29], [30]. A cluster is a group of interconnected nodes with a dedicated node called clusterhead. Clusterheads are responsible for cluster management, such as scheduling of the medium access, dissemination of control messages, or data aggregation. HEED by Younis and Fahmy [31] also targets QoS by using an efficient energy consumption method. Zhou et al. [32] obtain the energy dissipation structure and the optimum number of clusters in heterogeneous WSNs under a mathematical model, which provides guidance for clustering protocol design. LEACH by Heinzelman et al. [33] is a clustering-based communication protocol, which randomly rotates the cluster heads to evenly distribute the energy load among the sensor nodes in the network. Aslam et al. [34] presented a technique that periodically selects clusterheads according to two parameters, in which actors find an optimal geographical location with respect to their associated clusterheads. Chen et al. [15] proposed a decentralized clustering algorithm for target tracking, which assigns different roles to sensor nodes.

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 may occur in any other existing clustering algorithm for mobile ad hoc networks [35], [36], [37]. Thus clusterhead election procedure is obsolete. However, the network and the clustering algorithm must be designed in such a way that the actor node is always the most attractive clusterhead in its surrounding. Furthermore, the cluster structure permits multi-hop clusters. Only a few clustering algorithms allow multi-hop clusters, i.e., clusters where cluster members can potentially be several hops away from the clusterhead [38]. Since actor nodes are specially equipped nodes to aggregate and process data while delivering a long life-time, the number of actor nodes must be minimized. This property reduces the number of clusterheads required by the network. This is also important when actors cannot be deployed very close to each other due to restricted access to the environment. These distinct features of the application scenario require a novel network organization approach. Therefore we use an approach inspired by KHOPCA [39], which is a multi-hop clustering scheme consisting of a set of simple and easy-to-implement acting rules locally.

III. 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 [40]. Hence, SOFROP is divided into two phases: the first phase is concerned with the network organization, where an overlay network is formed and continuously adapted. The second phase is responsible for the data transmission. In this section, the system model is explained briefly and the two phases of SOFROP are described in detail.

A. System model

We consider a wireless actor and sensor network N with the number of nodes $|N| = n$. The wireless network N consists of a set of actor nodes A and a set of sensor nodes S, equipped with wireless communication capabilities. Our model also includes a sink node responsible for data aggregation and enabling connectivity to a backbone network. Each element in N has a transmission range r with a circular transmission area covering a total area of $\pi \cdot r^2$. 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 s_1 must be in the transmission range of s_2, i.e $d(s_1, s_2) \leq r_s$.

1) Sensor nodes: For each sensor node s in S, we assume 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 an update frequency f or triggered by an event.

Every node is able to communicate only with its current 1-hop neighbors (a sensor node or an actor), thus all communication in this model is locality preserving. Geographical positions of the nodes are assumed to be unknown. Since data is transmitted in only one direction and only local information is used, no multi-hop control communication is needed. 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 variable speed, acceleration and directions.

2) Actors and the sink: SOFROP manages actor-actor communication efficiently to save battery lifetime. The actor nodes use their full transmission range in two cases only. The first case 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 actor node has at least one actor or sink in its transmission range. The sink communicates only with actor nodes and it is also positioned in the transmission range of at least one actor node. 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 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 retransmitted with an incremented hop count only if its hop count is less than the actor's.

The second case is when the actor 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. Although actor nodes typically have stronger resources and more energy budget relative to sensor nodes, resource constraints apply to both sensors and actor nodes [12]. 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.

B. Network organization

The clustering is employed for the network organization in SOFROP. SOFROP must deal with the fact that actor nodes are pre-assigned clusterheads and are not supposed to change their status throughout the life-time of the network. Additionally, due to restrictions in the deployment of actor nodes, multi-hop clusters must be created as a remedy and the number of actor nodes must be minimized. The mobility of sensor nodes increases the number of re-affiliations to the actors.

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 most commonly used 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 due to the energy constraints of sensor nodes. To address the energy requirements of the Amazon scenario, we propose a different approach to transfer the weight information:

- Only the actor nodes generate packets periodically, from the start of the network lifetime to the end. These packets are called *Area Configuration Packets* (ACP). An ACP includes actor ID and hop value fields. The actor initializes these fields respectively with its ID and hop value.
- 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 loses its connection to the actors, it sets its hop value to the maximum hop value defined for the network. A node loses its connection when it doesn't receive an ACP (either directly from an actor or by retransmission of other nodes) for the predefined time defined for the network.

The nodes that lose the connection will be only in "listening" mode and they will not transmit any packets while actor nodes periodically send ACPs. This structure is suitable for WSANs since the complexity and resource requirement is focused on the actor nodes and it requires less energy than beaconing for the sensor nodes.

2) SOFROP clustering algorithm: The network structure in SOFROP is formed and maintained by the state transitioning rules of the clustering algorithm. Consider a node v with weight $w(v)$. The state transition for node v is given in Algorithm 1.

As each node applies Algorithm 1, the network structure is formed by copying the lowest neighbor weight increased by one as the sensor nodes move into the transmission range of the actors. This property is important in creation of a hierarchical structure. When a lower weight node, which is not an actor attracts surrounding nodes with higher weights, this node successively increases its weight to avoid a fragmented structure.

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

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

according to our network organization phase no logical communication path is built. In order to include the isolated nodes in the network, a solution could be to increase the difference of 0 and k, forcing this phase to built longer communication paths. The network designer, however, must consider the velocity and perturbations of the river affecting the nodes. If the paths become extremely long, no effective routing can be conducted or the messages from the most distant node may fail to 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 the decision making process and all the nodes rapidly update their data as the network structure changes. In SOFROP, sensor nodes use only the information on the packets they receive; they do not keep any global data about the network.

C. Data transmission

The network organization phase provides sensor nodes the information about lower hop neighbors and the number of hops needed to reach the closest actor. The sensor nodes collect information from the environment as they float in the river with the objective to transfer data to the actor nodes.

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 [41]. Sensor nodes must be aware of the sink's interests in order to gather the required information from the environment while they float in the river. One possible approach to this problem is conveying the interest information to sensor nodes as they float in the river. This method allows a sink to dynamically distribute or change its interests anytime in the network. However in our application scenario, the sensor nodes are mobile and they move with the flow of the river, creating random and hard-to-estimate paths. The current of Amazon River can reach up to a speed of 7 km/hr. Having this speed and mobility pattern, some sensor nodes will be accessible for very short periods of time, which may not be enough to convey the interests and collect data from the environment according to these interests. Moreover, a sensor node may observe important events before it receives the interests, which can result in data loss. This may be critical for the network since the sensor nodes do not follow repeating paths. Changing interest distribution also requires a high number of updates as the actor areas continuously change in SOFROP. Therefore conveying the interests is feasible in a more stable scenario, for instance when the nodes are stationary. In SOFROP, the interests are predefined at sensor nodes before they are thrown into the river. Each sensor has a predefined list of the information to be collected from the environment, called the "interest table".

The SOFROP's main goal for data transmission is the fair allocation of network resources among flows of different interests so that each flow is transmitted with a rate depending on its interest's priority. This goal can be achieved by a network if all the nodes in the network keep detailed state information for each flow and keep a record of the changes in the data for each flow periodically. However this approach would require high computation and memory resources for nodes, which is not suitable for WSANs. In SOFROP, 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. The receiving node takes the forwarding decision related to this packet based on the packet's information. We use eight bits to express the packet transmission rate and three bits to express the interest.

When an event is captured by a sensor node, the node checks its interest table to decide whether the sink needs to be notified of this event or not. If there is an interest for that event in the interest table, it is called an *on* interest for that node and the node generates data packets to report the event to the closest actor. A sensor node keeps the total number of *on* interests. The maximum packet transmission rate of a sensor node is called output capacity (C_o) in SOFROP. The rate field of the received packets is used to determine the remaining output capacity (C_r) of the node. C_o is reduced by the rate value on the first packet of an interest and recorded as the remaining output capacity (C_r) of the node.

Sensor nodes have buffers with predefined sizes. The buffer acts as a temporary space where the packets are held until the output link is available. Sensor buffers are simple in SOFROP; they work in first in first out (FIFO) fashion, outputting packets in the order they arrive. If we assume that the packet arrival process

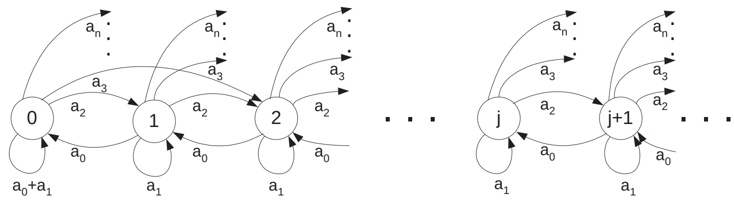

Fig. 2. State diagram of the Discrete Time Markov Chain for the queue length of a sensor node.

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 follows:

$$B_k = max(0, B_{k-1} + A_k - 1)$$

The underlying stochastic process of B_k can be described by a Discrete Time Markov Chain (DTMC) with states $q_i = P(N = i)$ [42]. The state diagram of the DTMC is shown in Fig. 2.

If the sensor node does not drop any packets for a period of time, it means $np_s \leq 1$ in one time window of this period. Then the steady-state of the number of packets in the buffer exists. Consequently the buffer occupancy can be formulated as follows:

$$B_k = max(0, B + A - 1)$$

Then its probability generating function (pgf) is found as follows:

$$
\begin{aligned}
G_B(z) &= \sum_{j=0}^{\infty} P(B=j)z^j \\
&= \sum_{j=0}^{\infty} q_j z^j \\
&= a_0q_0 + \sum_{j=0}^{\infty} P(B+A-1=j)z^j \\
&= a_0q_0 + \frac{G_B(z)G_A(z)-a_0q_0}{z} \\
&= \frac{a_0q_0(z-1)}{z-G_A(z)}
\end{aligned}
\tag{1}
$$

The probability generating function satisfies $G_B(1)=1$. Since $\lim_{z\to 1} a_0q_0(z-1) = \lim_{z\to 1} z - G_A(z)$ $=0$, we can apply l'Hopital's rule:

$$
1 = G_B(z) = \frac{a_0q_0}{1-G'_A(1)} = \frac{a_0q_0}{1-np_s}
$$

Therefore $a_0q_0 = 1-np_s$. After substitution,

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

The expected value of $G_B(z)$ is equal to the mean steady-state queue size of the buffer. It 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. This value is an important parameter for the algorithm used to forward/drop packets. The fair rate for a node depends on the number of *on* interests (N_i) and defined as follows:

$$
\alpha_f = C_o/N_i
$$

It is important to note that if all the packets are received with rates greater than α_f, α_f is the maximum rate value that a packet will be encoded with when C_r is negative. In other words, a packet is encoded with the minimum of its old tag value and the fair rate value before it is transmitted. However, this tagging is not very efficient when C_r is negative and there are flows with rate values lower than α_f. In such a case, if all the packets are encoded with rate values smaller than or equal to α_f, there will be an excess capacity not utilized at the sensor node. In order to utilize the excess capacity, a parameter called shared capacity (C_s) is defined. C_s is the output capacity shared among flows received with rates greater than α_f.

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. 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 in the node 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. The new α_p encoded on the packet is defined as follows:

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

where 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.

When a sensor node is not affiliated with an actor area, it waits for an area configuration packet (ACP). The node does not transmit any packets while waiting for an ACP, but fills its buffer with the packets it generated according to the sensed events. This is an efficient and feasible approach since the topology of the network changes continuously and the node can transmit the generated packets whenever it receives

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 packet's rate field
          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
```

an ACP from a sensor node. If the node cannot be affiliated to an actor area before its buffer becomes full, it keeps its buffer updated with the latest observations. Table I summarizes the important parameters used in our protocol.

D. Illustration of SOFROP

An example sequence for network organization of four sensor nodes and one actor is demonstrated in Fig. 3. At 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 network structure is formed.

TABLE I
ALGORITHM PARAMETERS

Number of "ON" interests	N_f
Number of greedy interest flows	N_g
Number of sharing interest flows	N_s
Output capacity	C_o
Remaining output capacity	C_r
Output capacity used by the sharing flows	C_s
Fair rate	R_e
Value of the rate field on a packet	R_p
Greedy flow flag	F_g
Sharing flow flag	F_s
Probability to drop a packet	P_d
Priority of an interest	ρ_i
Interest tag	i_p
Rate tag	α_p

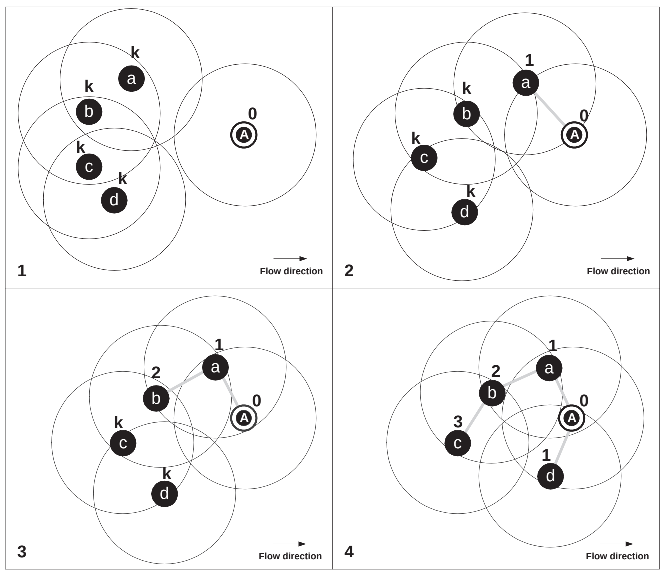

Fig. 3. An example sequence of a small group of sensor nodes and an actor.

When the sensor nodes get into transmission range of an actor, they start to take weights according to Algorithm 1.

An example of a lower weight node attracting surrounding nodes is illustrated in Fig. 4, where one of the nodes is moving faster compared to the other nodes in the scenario. This 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 one since it is directly connected to the actor at the beginning, but it loses its connection to the actor after it moves further away. However it still receives ACPs since it is in the transmission range of a node with weight three. 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. 5. Three actors are deployed uniformly at random and remain static while 60 sensor nodes are flowing from

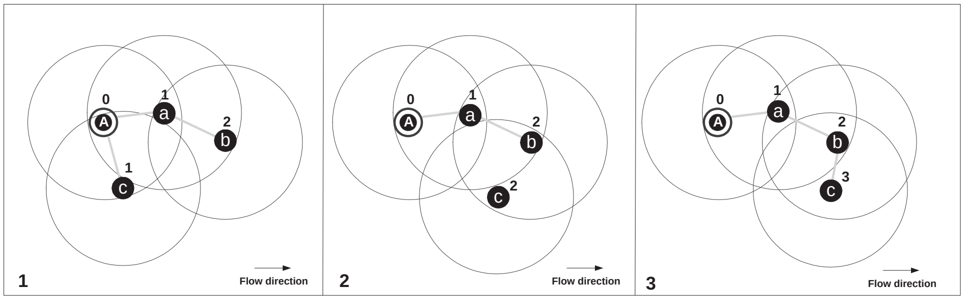

Fig. 4. An example sequence demonstrating the feature avoiding a fragmented structure in SOFROP.

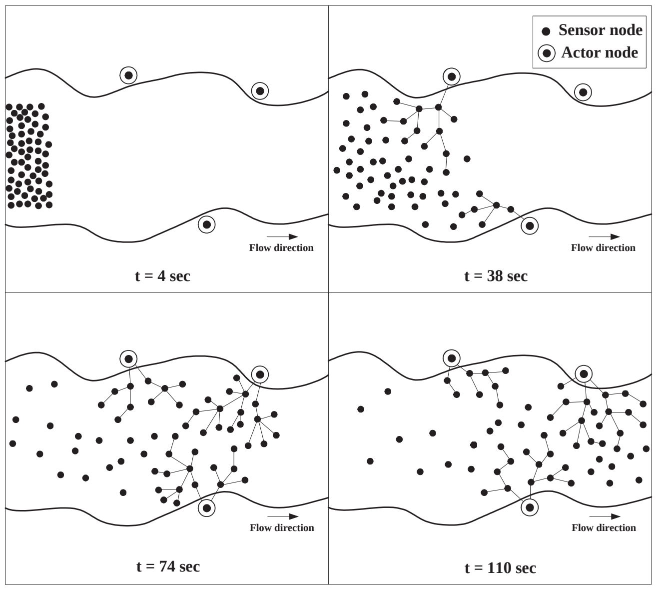

Fig. 5. A sequence of the dynamic overlay network produced by SOFROP network organization.

left to right, where the maximum hop-count allowed by the network organization is four. Note that Fig. 5 depicts only one of the different outcomes possible due to asynchrony.

An example sequence of packet transmissions in a system of four sensor nodes is shown in Fig. 6, 7 and 8. This system can be considered as a collection of any four sensor nodes in a network where SOFROP is employed. The initial state of the system is shown in Fig. 6. Nodes a and b have the hop-counts of three, node c and node d have hop-counts of two and one respectively. All sensor nodes are assumed to have an output capacity of ten packets per second. The packet type is presented in the packet header (T1: type-1 and T2: type-2). The value of the packet's rate field is shown with the numerical value and the destination of the packet is presented as a letter, showing the destination sensor node.

Sensor node a has a packet with the rate field value of eight and sensor node b has a packet with the rate field value of five. Both of these packets are destined for node c and they belong to different interests. An important property of SOFROP is denoted in Fig. 7. Since b is in transmission range of a, it receives

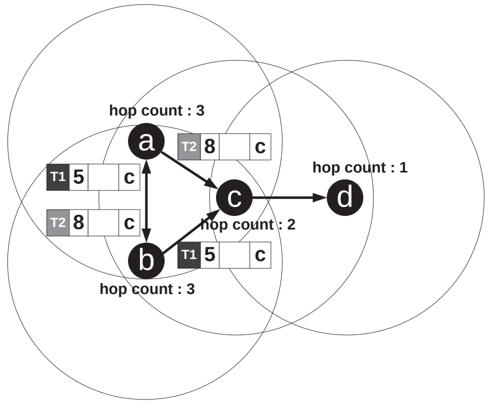

Fig. 6. An example system for demonstration of routing and labeling.

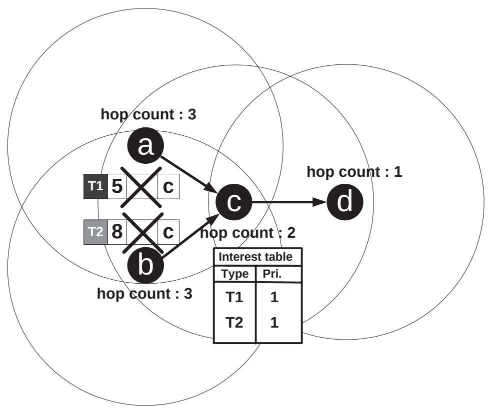

Fig. 7. An example system for demonstration of routing and labeling.

the packet from a. However the packet is not processed and directly ignored at b since the destinations for data at sensor nodes are determined during the network organization. The interest table of node c is also shown in Fig. 7. The types of packets from a and b have the same priority and these priority levels are predefined in the table of node c.

The node c receives the packets of two different types and these two flows share the output capacity of node c. Since both of the interests have the same priority, they must share the output capacity equally according to SOFROP. Since the output capacity of the node is ten packets per second, the fair share of the bandwidth for two flows is five packets per second. The node c drops packets from the flow, which are recived from node a. Therefore the rate values on the packets, which are received from a and forwarded to d, are updated according to the calculated P_d and α_p. In Fig. 8, the rate field of the forwarded packet, which was received from a, is assumed to be changed to five, which would be the case in ideal conditions.

In Fig. 9 and 10, the same system of nodes as in Fig. 6 is used with different values for the rate fields of the packets. The values of the rate fields are changed to six and two in this example for the packets

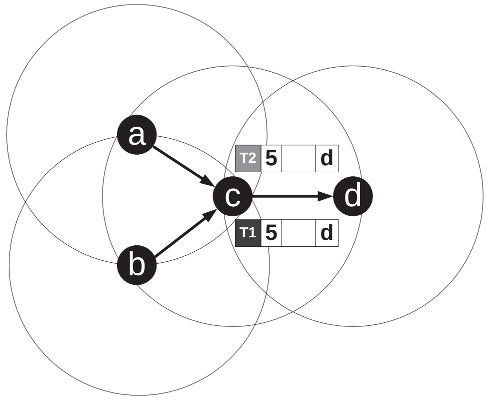

Fig. 8. An example system for demonstration of routing and labeling.

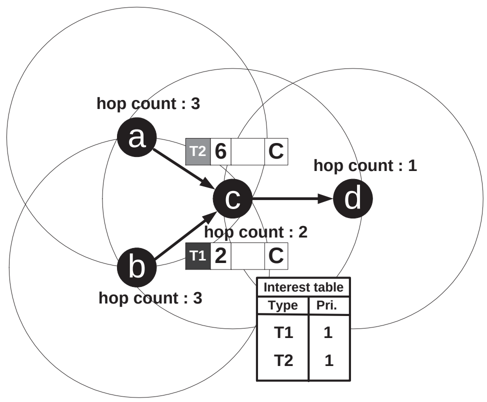

Fig. 9. An example system demonstrating bandwidth utilization.

coming from nodes a and b respectively. All packets have the same priority, which is predefined in the interest table of node c.

According to the initial property of SOFROP, which is described in Fig. 6 to 8, flows with equal priority share the output capacity of the transmitting node equally. Therefore if only that rule is employed by SOFROP, then the type-1 packets transmitted by node c will have rate values of two. This means most of the type-1 packets will be dropped according to P_d calculation although a big portion of output capacity of the node is not utilized. This is not acceptable in Amazon scenario where QoS has high priority as explained in the initial sections. Therefore in this scenario, packets of both flows are transmitted with unmodified rate fields.

In Fig. 11 and 12, the same system of nodes as in Fig. 6 is used with different rate fields of the packets to demonstrate that both packet priority and bandwidth utilization have important effects in the decision making process of SOFROP. In this example, the rate fields of the packets are changed to five and six. Different from the previous examples, the packets received from nodes a and b have different priorities.

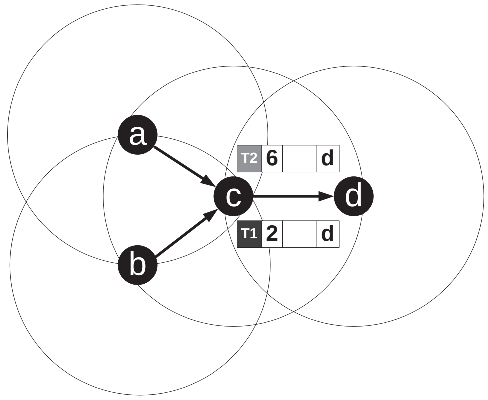

Fig. 10. Both of the packets are forwarded unmodified.

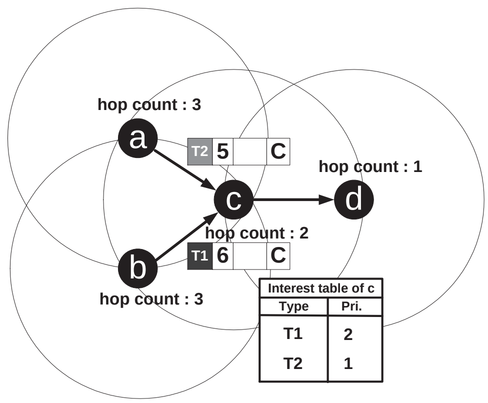

Fig. 11. An example system for demonstration of routing and labeling.

The priority of the packets from node b is twice the priority of the packets from node a.

The priorities of the flows are also important in this example when dropping the packets. Since there are two active flows on the node and the priority of one is half of the other, there will not be any packet drop from the flow with higher priority unless α_p becomes greater than $C_o \cdot 2/3$ according to SOFROP. Therefore packets from node b are forwarded without modification whereas the packets from node a are dropped with the corresponding P_d and the values of the rate fields of forwarded packets from node a are changed accordingly. In Fig. 12, the rate field of the forwarded packet, which was received from node a, is changed to four.

IV. Simulation Study

In this section, we perform extensive simulations to analyze the effectiveness of the proposed routing protocol.

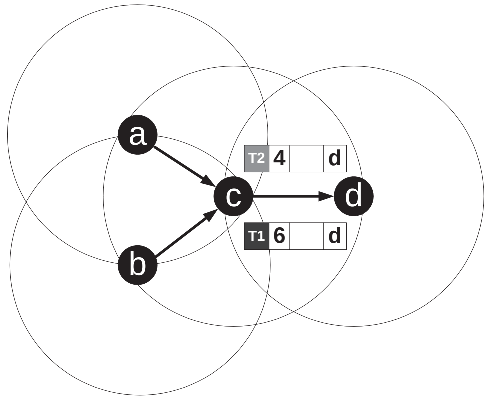

Fig. 12. An example system for demonstration of routing and labeling.

TABLE II
SIMULATION PARAMETERS

Number of sensor nodes	60
Total area	200x300
Sensor transmission range	40 (meters)
Number of actor nodes	4
Traffic type	CBR
Packet generation rate	10 packets/sec
Data packet size	256 bytes

A. Simulation environment and metrics

The simulations are carried out by OPNET modeler [43]. The transmission range of a sensor node is 40 meters, a realistic range for a sensor node (for instance Cerpa et al. [44] 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 queue size of 20 packets and a data rate of 10 packets per second. The IEEE 802.11 is used as the underlying MAC layer of the nodes and 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 II summarizes the simulation parameters used in our experimental setup.

The communication graph is built according to the system model specified in Section III. In each simulation, a network topology is generated with the sink located at one side of the area and actor located randomly either on the sides of the river or on the islands. The communication links may fail or disappear from the network caused by several reasons such as obstacles in the river. 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 III.

The protocol stack of the sensor node model is created in OPNET modeler as shown in Fig. 13.

TABLE III
MOBILITY SETTINGS

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

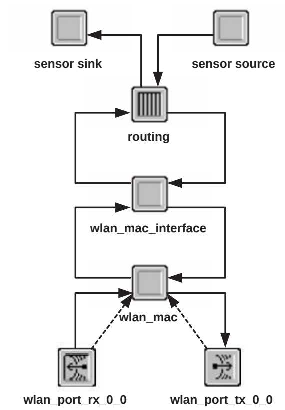

Fig. 13. Sensor node model created in OPNET.

Wireless local area network (WLAN) receiver and transmitter form the physical layer of a sensor node model. WLAN MAC layer interface is the data link layer used in OPNET 802.11 implementations and it is the interface between the routing layer and the WLAN MAC layer. The attributes of underlying IEEE 802.11 MAC layer used are shown in Table IV. The routing layer is where algorithm of SOFROP is mainly implemented.

Amazon River is the second largest river in the world with islands on it and its width ranges from a few hundred meters to 10 kilometers even at low season. The simulation study does not reflect actual dimensions of the Amazon scenario since we concentrate on the reproducibility of the results in the current work. As a part of the future work, we plan to conduct simulations with exact dimensions and more complex mobility models, and real world experiments on site.

TABLE IV
MAC LAYER ATTRIBUTES OF SENSOR NODES

Physical characteristics	Direct sequence
Transmit power (W)	$8.02 \cdot 10^{-6}$
Packet reception power threshold (dBm)	-95
Channel settings	Auto assigned
Short retry limit	7
Long retry limit	4
PCF	Disabled
HCF	Not supported

B. Simulation results

We study the effect of the proposed algorithm with the following simulation metrics: fairness, number of packets received, maximum hop value in the network and number of sensor and actor nodes.

1) Experiment 1: In order to create data traffic, twenty traffic sources are randomly placed in the network. These event sources simulate the points where it is possible to make observation for the sensor nodes in the network. They generate three different types of packets with constant rate of 10 packets per second, which creates congestion and bottlenecks in the network from time to time depending on the dynamic topology. All three of these traffic types have equal priorities. While 50% of the produced packets are type-1, type-2 and type-3 traffic have 30% and 20% allocations respectively. Twenty simulation runs were executed; the actor nodes and event sources are distributed randomly in the area for each simulation run. In order to see the effect of fairness, the same set of simulations is performed without using the fairness property of SOFROP. In other words, the rates or priority fields of the packets are not taken into account while taking routing decisions.

In Fig. 14, we can observe two important characteristics of SOFROP. First, the number of received packets for each type is very close to each other. Since packets from each type are produced at very large numbers, they create bottlenecks in the network. At these bottleneck points, higher the rate on a packet, greater the chance of that packet being dropped according to the routing principles of SOFROP. SOFROP drops more packets from the type of traffic with higher rate among the types with same priorities in a congestion situation. This is a desired property for the network since the sensor nodes are collecting information from the network and information on a single traffic type should not suppress the others. However when we take fairness properties out of SOFROP, we cannot observe the same property. Type-1 traffic receives more resources than for the other types in this case and additionally the total number of received packets is smaller.

2) Experiment 2: In order to test another property of SOFROP, the same set of simulations is run with different settings. The first setting change is in the percentages of the produced traffic types. In this experiment, 50 percent of the produced packets are type-1, 45 percent of them are type-2 and only 5 percent are type-3. As for the second change, we also include priorities in this case. While type-1 and type-2 packets have the same priority, the priority of type-3 packets is three times larger. This means that type-3 information is critical for the network. Fig. 15 shows the number of packets of each type received

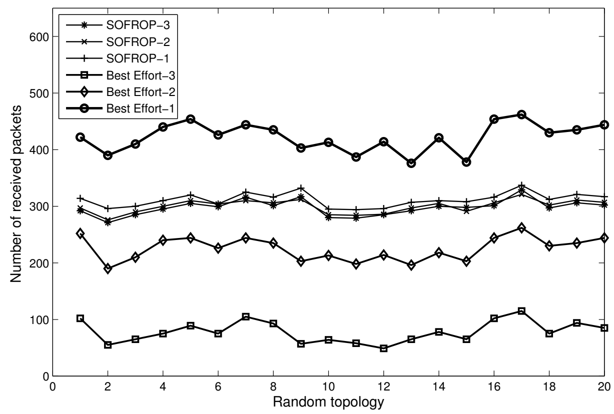

Fig. 14. Number of packets received by actors in Experiment 1.

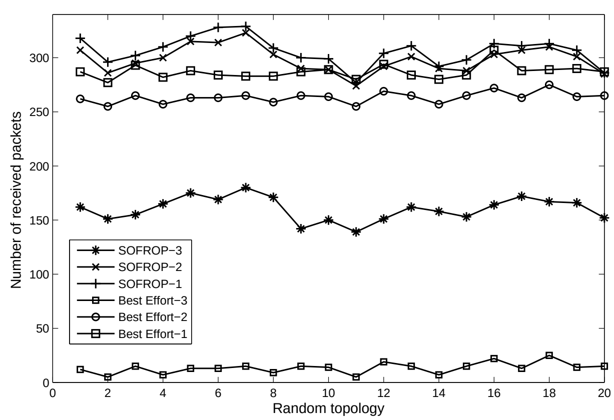

Fig. 15. Number of packets received by actors in Experiment 2.

by the actors. The results show that SOFROP protects the critical type of traffic and drops a very small number of packets.

3) Experiment 3: Another set of twenty simulations with the same settings in Experiment 1 is run without using the bandwidth utilization property of SOFROP. In Fig. 16, the lines corresponding to the runs with SOFROP are labeled as "SOFROP" and the lines corresponding to the runs without the utilization property are labeled as "No BW Util.". Therefore in these experiments, the only constraint is fairness but the utilization of the resources is not taken into account while taking routing decisions for "No Util." cases.

In Fig. 16 the number of received packets for each type of packets is very close to each other in both cases. This is the property observed in Fig. 14, which is also expected in the runs without utilization since the only constraint is fairness. However we also observe that the number of received packets by

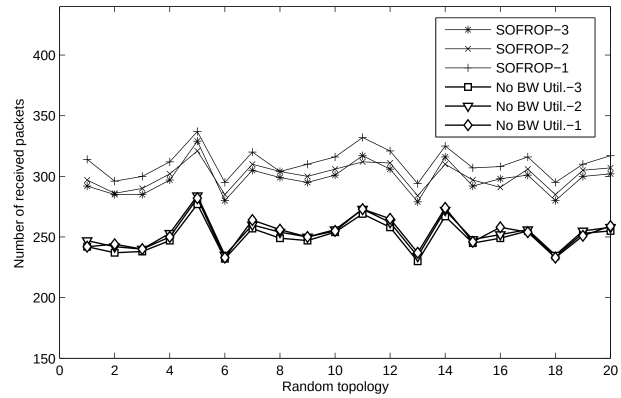

Fig. 16. Number of packets received by actors in Experiment 3.

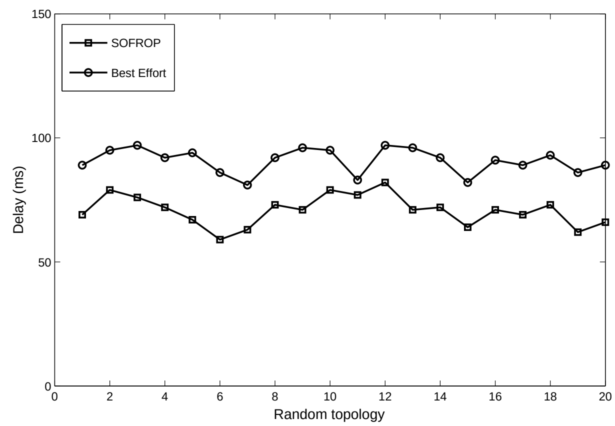

Fig. 17. End-to-end delay in Experiment 4.

actors without utilization property is less than SOFROP. The output capacity of each sensor node in the network is used at most three times the rate of the flow with the minimum rate since all flows have same priorities.

4) Experiment 4: The delay characteristics of SOFROP are observed by using a simulation set similar to the one 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. 17 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.

5) Experiment 5: The SOFROP's coverage properties are investigated using the same simulation settings as the previous experiment 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. The total of 25 simulations are run for each value of k, where 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. 18 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. 18 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.

V. 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 a few strategic locations close to an otherwise difficult to access river. The sensor nodes, smaller in size, are then deployed into the river to measure temperature, depth, pollution, flow speed, etc. The main characteristic of this scenario compared to others is that the actor nodes remain static but irregularly deployed while the sensor nodes are moving in an unpredictable pattern according to the river dynamics.

SOFROP is designed for this unique environment and consists of two phases: First, the network organization builds a structured network topology that permanently adapts according to the river dynamics. Second, the data transmission phase is responsible for data collection, aggregation and forwarding from the sensor to the actor nodes. We show that SOFROP provides fairness among different types of applications according to their priorities and rates. The packets carry their rate and values on their path to the actors, and they are dropped probabilistically at the sensor nodes. SOFROP also utilizes the excess bandwidth with a

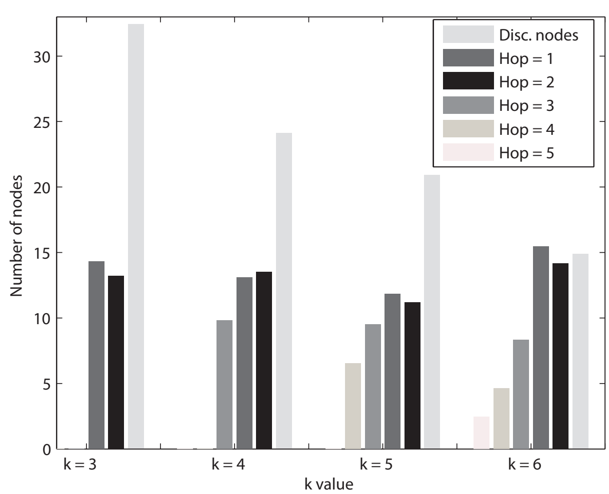

Fig. 18. Number of sensor nodes and their hop values for different k values.

lightweight algorithm in the sensor nodes and uses a buffer at each sensor node for the observations made while the node is not affiliated with an actor area. An additional property of SOFROP is its scalability achieved by a fully local approach. These features make SOFROP an ideal routing protocol for wireless networks with mobile sensors and stationary actors. Simulation results verify the effectiveness of the proposed scheme in the unprecedented conditions of the chosen scenario.

VI. Acknowledgement

The authors would like to thank OPNET Technologies Inc. for supporting this research by providing OPNET Modeler software under OPNET University Program. Matthias R. Brust is grateful to FAPESP (Grant no. 2008/52129-9).

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, August 2008.

[3] T. Naumowicz, R. Freeman, H. Kirk, B. Dean, M. Calsyn, A. Liers, A. Braendle, T. Guilford, and J. H. Schiller, "Wireless sensor network for habitat monitoring on Skomer Island," in *Proc. of the IEEE Conference on Local Computer Networks (LCN)*, October 2010, pp. 882–889.

[4] T. Ma, M. Hempel, K. Hua, D. Peng, and H. Sharif, "A novel cooperative image transmission scheme in wireless sensor networks," in *Proc. of the IEEE Conference on Local Computer Networks (LCN)*, October 2010, pp. 240–243.

[5] S.-F. Li, "Wireless sensor actuator network for light monitoring and control application," in *Proceedings of the Consumer Communications and Networking Conference, (CCNC)*, January 2006, pp. 974–978.

[6] T. Melodia, D. Pompili, V. C. Gungor, and I. F. Akyildiz, "A distributed coordination framework for wireless sensor and actor networks," in *Proceedings of MobiHoc*, May 2005, pp. 99–110.

[7] 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.

[8] J. Al-Karaki and A. Kamal, "Routing techniques in wireless sensor networks: A survey," *Wireless Communications*, vol. 11, no. 6, pp. 6–28, December 2004.

[9] 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 *Proceedings of the IEEE Conference on Local Computer Networks (LCN)*, December 2009, pp. 819–825.

[10] D. Peleg, *Distributed computing: A locality-sensitive approach*. Society for Industrial and Applied Mathematics, 2000.

[11] J. Nagle, "On packet switches with infinite storage," *IEEE Transactions on Communications*, vol. 36, no. 4, pp. 435–438, April 1987.

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

[13] X. Masip-Bruin, M. Yannuzzi, J. Domingo-Pascual, A. Fonte, M. Curado, E. Monteiro, F. Kuipers, P. V. Mieghem, S. Avallone, G. Ventre, P. Aranda-Gutirrez, M. Hollick, R. Steinmetz, L. Iannone, and K. Salamatian, "Research challenges in QoS routing," *Computer Communications*, vol. 29, no. 5, pp. 563 – 581, March 2006.

[14] M. A. El-Gendy, A. Bose, and K. G. Shin, "Evolution of the Internet QoS and support for soft real-time applications," in *Proceedings of the IEEE*, vol. 91, no. 7, July 2003, pp. 1086–1104.

[15] D. Chen and P. K. Varshney, "QoS support in wireless sensor networks: A survey," in *Proceedings of the 2004 International Conference on Wireless Networks (ICWN 2004), Las Vegas, Nevada, USA*, June 2004.

[16] Y. Li, C. S. Chen, Y.-Q. Song, and Z. Wang, "Real-time QoS support in wireless sensor networks: A survey," in *International Conference on Fieldbuses and Networks in Industrial and Embedded Systems*, November 2007, pp. 159–164.

[17] W. Hu, N. Bulusu, and S. Jha, "A communication paradigm for hybrid sensor/actuator networks," in *Proceedings of PIMRC*, September 2004, pp. 201–205.

[18] G. A. Shah, M. Bozyigit, O. B. Akan, and B. Baykal, "Real-time coordination and routing in wireless sensor and actor networks," in *Proceedings of Next generation teletraffic and wired/wireless advanced International Conf. (NEW2AN)*, May 2006, pp. 365–383.

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

[20] K. Morita, K. Ozaki, N., T. Enokido, and M. Takizawa, "Evaluation of reliable data transmission protocol in wireless sensor-actuator network," in *Proceedings of Advanced Information Networking and Applications Workshops (AINAW)*, vol. 2, May 2007, pp. 713–718.

[21] A. Boukerche, R. B. Araujo, and L. Villas, "A novel QoS based routing protocol for wireless actor and sensor networks," in *Proceedings of GLOBECOM*, November 2007, pp. 4931–4935.

[22] A. Boukerche, R. Araujo, F. Silva, and L. Villas, "Wireless sensor and actor networks context interpretation for the emergency preparedness class of applications," *Computer Communications*, vol. 30, no. 13, pp. 2593–2602, September 2007.
[23] V. Paruchuri, A. Durresi, and L. Barolli, "Energy aware routing protocol for heterogeneouswireless sensor networks," in *Proceedings of Database and Expert Systems Applications*, August 2005, pp. 133–137.
[24] 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, February 2009.
[25] T. Houngbadji and S. Pierre, "QoSNET: An integrated QoS network for routing protocols in large scale wireless sensor networks," *Computer Communications*, vol. 33, no. 11, pp. 1334–1342, July 2010.
[26] M. Baunach, "Collaborative memory management for reactive sensor/actor systems," in *Proc. of the IEEE Conference on Local Computer Networks (LCN)*, October 2010, pp. 953–960.
[27] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2826–2841, October 2007.
[28] W. Heinzelman and A. C. snd H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, October 2002.
[29] K. Dasgupta, K. Kalpakis, and P. Namjoshi, "An efficient clustering-based heuristic for data gathering and aggregation in sensor networks," in *Proceedings of WCNC*, vol. 3, March 2003, pp. 1948–1953.
[30] 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.
[31] O. Younis and S. Fahmy, "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks approach," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, Oct.-Dec. 2004.
[32] H. Zhou, Y. Wu, Y. Hu, and G. Xie, "A novel stable selection and reliable transmission protocol for clustered heterogeneous wireless sensor networks," *Computer Communications*, vol. 33, no. 15, pp. 1843–1849, September 2010.
[33] W. R. Heinzelman, A. Ch, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *System Sciences '00: Proceedings of the 33rd Annual Hawaii International Conference on*, 2000, pp. 3005–3014.
[34] N. Aslam, W. Phillips, W. Robertson, and S. Sivakumar, "Extending network life by using mobile actors in cluster-based wireless sensor and actor networks," *IFIP, Wireless Sensor and Actor Networks*, vol. 264, pp. 74–84, May 2008.
[35] M. Chatterjee, S. Das, and D. Turgut, "WCA: A weighted clustering algorithm for mobile ad hoc networks," *Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks)*, vol. 5, no. 2, pp. 193–204, April 2002.
[36] S. Basagni and E. Jonsson, "Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks," in *In Proceedings of International Vehicular Technology Conference, VTC*, September 1999, pp. 889–893.
[37] N. Aydin, F. Nait-Abdesselam, V. Pryyma, and D. Turgut., "Overlapping clusters algorithm in ad hoc networks," in *Proceedings of the IEEE GLOBECOM'10*, December 2010.
[38] O. Dousse, P. Thiran, and M. Hasler, "Connectivity in ad hoc and hybrid networks," in *INFOCOM*, November 2002, pp. 1079–1088.
[39] M. R. Brust, H. Frey, and S. Rothkugel, "Adaptive multi-hop clustering in mobile networks," in *Proceedings of the 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.
[40] N. Burri, P. von Rickenbach, R. Wattenhofer, and Y. Weber, "Topology control made practical: Increasing the performance of source routing," *Lecture Notes in Computer Science*, vol. 4325, no. 2, pp. 1–12, 2006.
[41] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *MobiCom*, August 2000, pp. 56–67.
[42] K. S. Trivedi, *Probability and statistics with reliability, queuing and computer science applications*. Chichester, UK: John Wiley and Sons Ltd., 2001.
[43] "Opnet modeler," http://www.opnet.com.
[44] 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.