

Lightweight Routing with Dynamic Interests in Wireless Sensor and Actor Networks

Mustafa İlhan Akbaş and Damla Turgut

Department of Electrical Engineering and Computer Science

University of Central Florida

Orlando, Florida 32816-2362

Email: {miakbas,turgut}@eecs.ucf.edu

Abstract

Wireless sensor and actor networks (WSANs) have been increasingly popular for environmental monitoring applications in the last decade. While the deployment of sensor nodes enables a fine granularity of data collection, resource-rich actor nodes provide further evaluation of the information and reaction. Quality of service (QoS) and routing solutions for WSANs are challenging compared to traditional networks because of the limited node resources. WSANs also have different QoS requirements than wireless sensor networks (WSNs) since actors and sensor nodes have distinct resource constraints.

In this paper, we present, LRP-QS, a lightweight routing protocol with dynamic interests and QoS support for WSANs. LRP-QS provides QoS by differentiating the rates among different types of interests with dynamic packet tagging at sensor nodes and per flow management at actor nodes. The interests, which define the types of events to observe, are distributed in the network. The weights of the interests are determined dynamically by using a nonsensitive ranking algorithm depending on the variation in the observed values of data collected in response to interests. Our simulation studies show that the proposed protocol provides a higher packet delivery ratio and a lower memory consumption than the existing state of the art protocols.

I. INTRODUCTION

Wireless sensor networks (WSNs) [1], which have been studied for more than a decade now, are designed to gather information about their environments. In most WSN applications, sensor nodes transmit the collected information to predetermined data collector nodes or users in the network. Sensor nodes are designed to be small and cheap: thus, in general have limited data processing capabilities, low transmission rates, small batteries, and limited memory capacity. There are, however, applications, where it is advantageous to add to the network *actor nodes* with better computation and communication capabilities, larger memories and long lasting batteries. These nodes are able to evaluate the collected information or execute computationally complex operations in the environment. Networks where sensor and actor nodes coexist are called *wireless sensor and actor networks* (WSANs) [2].

WSANs have a heterogeneous structure in terms of node types and resources. Any protocol, which aims to provide quality of service (QoS) to the users of the WSAN, must exploit this heterogeneity by modifying its routing strategies such that they take advantage of the higher resources provided by the actor nodes. Therefore, WSANs cannot be simply regarded as WSNs when aiming for QoS support in the network.

Throughout this paper, we will assume that the information to collect from the environment depends on the stated *interests* of the users. Different interests have different QoS requirements. Let us consider a fire alarm system, which provides information directly to fire fighters. Information about a sharp temperature increase is high priority and must be transmitted by all the nodes in the system without regards to the resources. The periodic reporting of normal temperatures on the other hand is less critical. Consequently, QoS support becomes a vital part of WSANs and it is widely used in various WSAN applications (see [3]).

In most of the WSAN applications, the interests and their requirements are predefined and remain constant during the lifetime of the network. However, the importance of the events observed by the sensor nodes in response to interests changes dynamically with the changing conditions of the network. This is important particularly for real-time applications, which must meet various constraints under resource limitations in dynamic environments. The network resources must be utilized efficiently to avoid cases such as keeping an initially important interest prioritized while the observed values for that interest stay constant.

In this paper, we propose a lightweight routing protocol with QoS support, which provides the highest rate available for each flow and dynamically prioritizes the interests according to the observed values in a WSAN. We assume a WSAN consisting of randomly distributed, stationary sensor and actor nodes. Actor nodes have no initial information on the positions of the sensor nodes. The network is formed among actors and the sink and it is defined as the “communication backbone”. Since the number of actors is low compared to the number of sensor nodes, some of the sensor nodes are not directly connected to an actor. These nodes perform hop-by-hop communication along paths determined by the interests.

In our approach, QoS is defined as the assurance of services required by the applications. The main optimization objectives are the packet transmission and loss rates. LRP-QS, a lightweight routing protocol with dynamic interests and QoS support, is developed to exploit the heterogeneous node structure of WSANs. At sensor nodes, it is lightweight and efficient in terms of memory and power consumption as opposed to actors which have a more complicated task in accordance with their capabilities. The interests of the sink are transmitted to the actors through the communication backbone with initial weights, which define their rate requirements. According to the events observed, these interests are ranked at each actor to adjust their weights according to the network conditions. The application scenario also includes bursty data sources resulting in congestion at various points in the network from time to time, which increases the criticality of efficient resource utilization in the network.

The rest of the paper is organized as follows. Section II summarizes the related work. We provide a detailed description for our approach in Section III. We show the simulation results in Section IV and finally conclude in Section V.

II. RELATED WORK

There are numerous approaches proposed for clustering and backbone formation in ad hoc networks [4]. The cluster heads have more energy and computation power compared to the regular nodes in most of these protocols, similar to the actors in WSANs (see [5], [6], [7], [8], [9]). The design of the routing schemes proposed for clustered networks is based on the selection of cluster heads and the network structure. The network organization of our protocol shares several aspects with cluster-based routing schemes. The cluster head selection approach by Soro and Heinzelman [6] favor nodes deployed in densely populated network areas to maintain the full network coverage. Smaragdakis et al. [10] propose stable election protocol (SEP), a heterogeneous-aware protocol to prolong the time interval before the death of the first node in a stationary network. For the election of cluster head, SEP requires the energy levels of all nodes, which is used as the metric for heterogeneity. Aslam et al. [11] finds optimal geographical locations for actor nodes with respect to their associated cluster heads. Zhang et al. [12] also use geographical locations of nodes and select cluster heads in adjacent groups of nodes close to one another. This property of the network is used to reduce the average energy consumption in the WSN. Chen et al. [13] introduce a dynamic clustering algorithm for target tracking, which creates a hierarchical network structure. Clustering Patch Hierarchical Routing Protocol (CPHRP) by Lin and Liao [14] uses network coverage rate and effective network lifetime to evaluate a WSN. The main objective of CPHRP is to optimize network coverage rate through clustering patch by using these metrics and hierarchical multi-path tree routing. The clustering routing protocol by Boukerche et al. [15] aims at optimizing energy dissipation in the network while providing fault tolerance. The algorithm alternates the inter-cluster communication nodes and possible routes to the sink to reduce energy expenditure. These protocols mostly focus on cluster head selection

and serve well for various applications. However our network organization is designed to be lightweight particularly at sensor nodes considering the application scenario with predetermined cluster heads. The application scenario also requires a structure with multi-hop transmission of collected information, in which cluster members can potentially be several hops away from the cluster head.

Support for efficient resource utilization is an important part of WSAN routing protocols. The routing paradigm by Hu et al. [16] builds an anycast tree rooted at each event source to reduce latency and energy consumption in communication. The dynamic behaviors of the sinks for joining or leaving the system shape these anycast trees. Cañete et al. [17] present a component-based framework, which combines macro-programming and node-centric programming to develop applications over WSANs with the specification of real-time constraints between services. Tao et al. [18] propose the flow-balanced routing (FBR) protocol, which aims to achieve power efficiency and coverage preservation. FBR assigns the transferred data over multiple paths from the sensor nodes to the sink to equalize the power consumption of nodes. Melodia et al. [19] use an event-driven clustering paradigm to design a sensor-actor coordination model and formulate the actor coordination as a task assignment optimization problem. The real-time routing framework by Shah et al. [20] addresses the coordination of sensor and actor nodes through the delay bound for distributed routing. Another coordination algorithm among actors is introduced with the real-time communication framework by Ngai et al. [21], where an event reporting algorithm for sensor-actor communication is also given to minimize the transmission delay. Ad Hoc On Demand Delay Constrained Distance Vector Routing (AOD²V) by Sama and Akkaya [22] also uses delay as the main constraint. AOD²V uses delay-EDD at admission control and EDF is used to determine the departure order of the packets at the intermediate nodes. The routing algorithm by Hung et al. [23] determines the maximum amount of data each node can transmit by taking energy as the main parameter. The data transmission protocol by Morita et al. [24] thrives to enhance reliability with redundancy. Xia et al. [3] apply feedback control for dynamic bandwidth allocation, which uses deadline miss ratio control to improve QoS in terms of reliability. Paruchi et al. [25] 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. Energy-balanced routing method based on forward-aware factor (FAF-EBRM) by Zhang et al. [26] uses the link weight and forward energy density to select the next-hop node. FAF-EBRM also uses a spontaneous reconstruction mechanism for local topology to balance the energy. In majority of these algorithms, the main objective is the energy efficiency at the sensor nodes. Directed Diffusion (DD) by Intanagonwiwat et al. [27] achieves energy savings by selecting empirically good paths and by data aggregation. The sink in DD expresses a set of interests regarding the information to be collected and each node records the neighboring node from which the interest is received. LRP-QS also uses interests for defining the information to be collected. However both the interests and the paths in LRP-QS are dynamically modified according to the changes in the observed data.

For efficient resource utilization, LRP-QS utilizes the distinct or changing requirements of the interests. QoSNET by Houngbadji and Pierre [28] 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. InRout [29] by Villaverde et al., a QoS aware route selection algorithm for WSNs, enables distributed route selection by sharing the local information among neighbouring nodes and using Q-learning techniques. The management protocol for reactive sensor and actor systems by Baunach [30] focuses on memory and offers a collaborative approach. Grid-based Multipath with Congestion Avoidance Routing protocol (GMCAR) [31] divides the network into grids and selects a master node in each grid. These master nodes route packets using the grid densities and the hop count. Energy Efficient and QoS aware multipath routing protocol (EQSR) by Ben-Othman and Yahya [32] maximizes the network lifetime by balancing energy consumption across a set of available paths in the network. The metrics used by EQSR to predict the next hop are residual energy, node available buffer size, and Signal-to-Noise Ratio (SNR). EQSR also employs a queuing model to handle real-time and non-real-time traffic through service differentiation. Hammoudeh and Newman [33] present cluster-based Route Optimisation and Load-balancing protocol (ROL) and . an algorithm for load balancing, Nutrient-flow-based Distributed Clustering (NDC). ROL/NDC uses a

combination of routing metrics, which are configured according to the priorities of user-level applications, to improve the network lifetime. Boukerche et al. [34] proposed “QoS Based Routing Protocol” (QBRP), a protocol with service differentiation, in which routes are generated at the actors by using the information collected from the sensor nodes. The next hop for each type of packet is forwarded to the nodes based on the data, which is centrally processed at the actor nodes. We proposed a routing protocol [35] using a similar approach as QBRP in network organization. LRP-QS extends this approach and eliminates heavyweight requirements of QBRP such as central processing and the need for frequent updates.

According to the desired characteristics of LRP-QS, a ranking method is used to compare the performances of different types of interest traffic in terms of the changes in values. This behavior is similar to the competition of teams in any sporting event. Therefore, ranking methods in sports can be employed to rank the interests. Colley [36] and Massey [37] are two of the most well-known linear algebra-based sports ranking methods with elegant formulations. Massey’s model depends on the rule that the difference in the ratings of two teams represents the point differential in a matchup of these two teams. Different than Colley method, Massey utilizes actual game scores and homefield advantage. There are also other methods make use of Markov chains ([38],[39]) in sports ranking. In a recent study, Chartier et al. [40] analyzed Colley, Massey and Markov methods in terms of sensitivity. Our ranking method at actor nodes is selected based on the result of this analysis.

III. ROUTING WITH QOS SUPPORT AND DYNAMIC INTERESTS

We are considering a network model composed of three types of nodes: sensor nodes, actor nodes and the sink node. All nodes in the network are stationary and the information to be collected is expressed with a set of *interests* by the sink regarding the information to be collected. The interests are stored in the sensor and actor nodes and they define the observations made, as well as the rates of the flows and employment of the transmission bandwidth.

A. Actor areas

The bootstrapping of the network configuration starts with the flooding of area configuration packets (ACP) by the actor nodes. An ACP includes the actor’s ID and the number of hops the packet is forwarded. Sensor nodes have *actor ID* and *hop value* attributes and neither of them has initial values. In other words, initially a sensor node knows neither the actor node it is associated with nor the hop-distance to that actor node.

When a sensor node receives an ACP, it first checks the hop value. If the value is greater than or equal to the node’s hop value, the packet is dropped. Otherwise the node updates its attributes with the values in the related fields of the packet and retransmits the packet. Thus, the sensor node keeps the information for only one actor node even when it receives ACPS from multiple actor nodes. The actor node will be the destination for its data packets. The node can keep multiple destinations and this record can be leveraged to have more control on resources. However one of the main objectives is to keep the protocol as lightweight as possible on sensor nodes. In addition, the hop value of a sensor node automatically changes when one of its neighbors fails since all communication is handled locally.

Once the actor areas are formed, each sensor node is affiliated with an actor, has information about its lower-hop neighbor(s) and the number of hops required to reach its actor. Figure 1 shows the hypothetical view of the network with the connections after this phase.

B. The communication backbone

The actor areas must be connected to transmit the collected data to the sink. We define the network formed by the links among actors and the sink as the *communication backbone*. The formation of the communication backbone is started by the sink sending an area integration packet (AIP) with its ID in the source field.

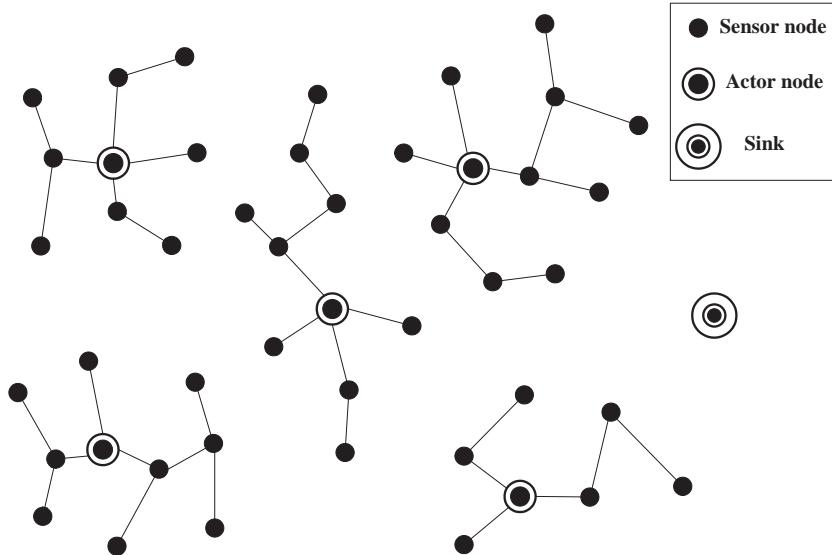


Fig. 1. The network view after formation of actor areas.

The destination address for data packets (DA_d) is an attribute of each actor. When an actor receives an AIP from a sink, it saves the sink as the destination address for data packets (DA_d). Then the AIP is forwarded in the network among actors. The sink must be positioned in the transmission range of at least one actor in order to prevent bottlenecks at the links close to the sink. Otherwise the sink receives the collected data through sensor nodes, which would create severe packet loss and delay.

If an actor doesn't receive an AIP from a sink, the first actor from which it received the AIP is recorded as the DA_d . Then the actor places its ID on the AIP and retransmits it. If an actor receives AIPs from multiple actors, it saves the extra actor IDs in the *redundancy list* (L_r). This list is kept at an actor for future use in case of a change in the communication backbone such as a dead actor node. The pseudocode of the algorithm used at each actor node receiving an AIP is presented in Algorithm 1.

If each actor is guaranteed to be in the transmission range of at least one actor, i.e. the network communication backbone is connected, then the AIPs do not need to be processed at sensor nodes. Figure 2 shows the network view with actor areas and a connected communication backbone.

Algorithm 1 Processing of AIP at an actor node

```

1: Check source address of AIP
2: Check  $DA_d$  attribute
3: if  $DA_d$  is a sink then
4:   drop AIP
5: else if AIP is received from a sink then
6:    $DA_d$  = ID of the sink transmitted AIP
7:   transmit AIP
8: else if AIP is received from an actor then
9:   if  $DA_d$  is an actor then
10:    Update  $L_r$  with the actor that transmitted AIP
11:   else
12:     $DA_d$  = ID of the actor transmitted AIP
13:    transmit AIP
14:   end if
15: else if AIP is received from a sensor node then
16:   if  $DA_d$  is an actor then
17:     drop AIP
18:   else
19:      $DA_d$  = ID of the actor on AIP
20:     transmit AIP
21:   end if
22: end if
  
```

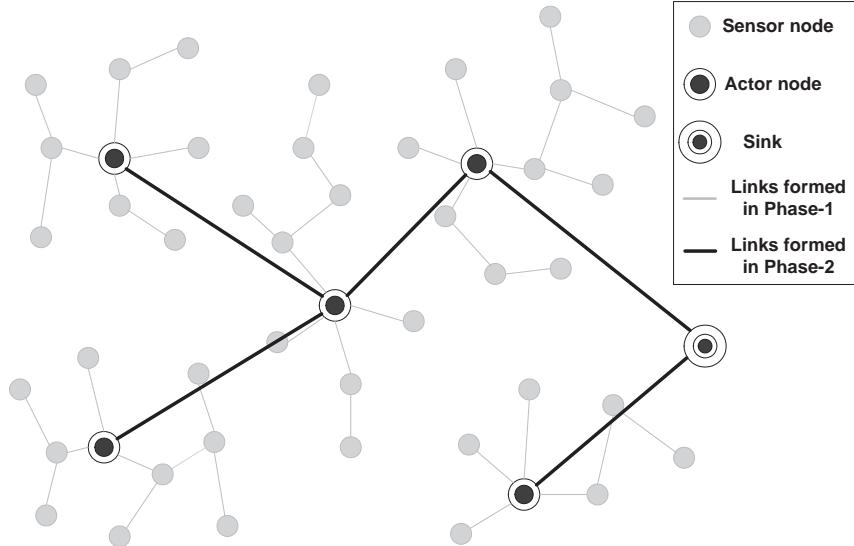


Fig. 2. Network view after actor areas and network among actors are formed.

When there is no actor node in the transmission range of an actor node, sensor nodes, which receive AIP of this actor node, use a lightweight algorithm to process these packets. The receiving sensor node checks if the actor ID on AIP is equal to the node's associated actor node ID. The packets with matching actor node ID are transmitted to the sensor node's higher-hop neighbor(s) and the first AIP from another actor is transmitted to the lower-hop neighbors in order to guarantee the conveying of AIPs in the network. All other AIPs are dropped by the sensor nodes.

C. Interest subscription

The interests are distributed to the actors via the communication backbone. When an actor receives an interest from the sink, it checks and updates its interest subscription table with the received interest.

Each actor transmits the interests to the closest sensor nodes to start the selective flooding (i.e. forwarding only the packets from higher-hop neighbors) of the interests in its area. When a sensor node receives a new interest, it updates its subscription table. If the sensor node observes an event the sink is interested, then it will generate data packets. However if a node does not sense an event but it is only on the path of data transmission, then it will keep the interest in its subscription table and use this information when routing the data packets.

At the end of this phase, each sensor node knows which type of packets to generate and which events to generate packets for. We define the interests, which the sensor nodes generate packets for, as the *active interests*. Each sensor node also has the interests in their subscription table for which they reside on the path from the reporting sensor node to the actor. This type of interests is defined as *passive interests*, for which sensor nodes keep only an on-off information.

D. Ranking interests

The requirements of users are distributed in the network by using methods based on the concept of interests. The term, interest, is introduced by Intanagonwiwat et al. [27] and in their system, interests are changed only when they expire or when they are explicitly overwritten by new interests. Our approach introduces a new and dynamic utilization of interests, where their relative importance can be determined locally by a self-organizing protocol. The events observed by the nodes form not only the basis of the packets forwarded to the sink, but they are also evaluated locally to adjust the importance ranking of the interests. Interests with higher importance ranking are allowed more resources, resulting in higher quality of observation, reduced traffic and less energy expenditure.

The importance ranking method of our approach is based on the amount of changes in the observed values for events of interest. For instance if the packets received in response to interest *A* in a predefined period of time have high fluctuations in their values compared to the constant values of interest *B* packets, then *A* is considered as more important than *B* *for the given time period*. The rankings of interests will be different at each actor area as the interests are ranked depending on the local events.

Finding an appropriate ranking algorithm is critical to the efficiency of LRP-QS. Let us start by outlining the requirements which must be satisfied by the ranking algorithms. The ranking algorithm must not be sensitive to small changes. This requirement eliminates a number of popular ranking algorithms from consideration. Let us consider a value which had been unchanged for a long period of time, and thus, it has a low importance ranking. If the value fluctuates for a single time period it should not yield a sudden increase to high importance.

Ranking algorithms had been studied in a number of contexts. Examples include sports ranking algorithms which rank sport teams based on their game winnings and web page ranking algorithms which rank the web pages with respect to a query based on a number of signals.

Chartier et al. [40] analyzed the sensitivity of the Colley, Massey and Markov methods to small perturbations and determined how much the ranking is affected by these changes. All of these well-known methods are based on linear algebra and have a formulation with a closed-form solution.

According to the analysis of Chartier et al. [40], similar discrepancies in the input and output ranking data showed instability of the ranking methods. Analysis results showed that while Colley and Massey methods are insensitive to small changes, Page Rank method is highly sensitive to such changes. Therefore, Colley and Massey methods are popular in sports ranking. For instance, these methods are two of the six computer methods used by the Bowl Championship Series to rank NCAA football teams. If Page Rank method is applied to sports ranking, small changes result in anomalies in the rankings. A single defeat in a sports game should not change the whole ranking list or result in a rank jump. The Colley and Massey methods have an isolated response, resulting in changes to the rankings of only two objects in question.

Colley method is chosen to be used in LRP-QS over the Massey method as our ranking application. Colley method is based only on results from the field. Massey method on the other hand employs metrics such as homefield advantage or scores which don't have any correspondence in our approach. Therefore, Colley method is indeed more appropriate for ranking interests compared to Massey method.

The Colley Matrix method [36] is used to produce fair rankings in sports based exclusively on the wins and losses of the teams. In the following, we describe the mathematical foundations of the Colley method, while also adapting its sports-based language to the language of interest importance ranking used in our algorithm. The interest i wins over interest j if its percentage change is higher. We denote with $n_{w,i}$ the number of wins of interest i . $n_{total,i}$ is the total number of comparisons made, which corresponds to the number of games in sports. In traditional ranking methods, the winning ratio, which is defined as $\frac{n_{w,i}}{n_{total,i}}$, is used. Instead of this, the Colley method defines a modified rating r_i for interests, shown as follows [41]:

$$r_i = \frac{1 + n_{w,i}}{2 + n_{total,i}}$$

The advantage of using this equation for the definition is to avoid the inconsistency in comparison of interests in specific conditions. For instance, the interests with no changes or the interests with changes in every *time period* would have values of 0 and 1 respectively if traditional win ratio is used instead of defined r_i . In our application, the interests can have initial values or the weights of the interests can be initially zero and dynamically modifiable. When interests have no initial weights after the first *time period*, the interest with a change is “infinitely better” than the interest without a change. Using the Colley method, the former interest ($r = \frac{2}{3}$) would have a twice the better score compared to the later interest ($r = \frac{1}{3}$). Hence, the initial rating of any interest with no changes is equal to $\frac{1}{2}$, which is the median value between 0 and 1. Depending on the comparisons, a win increases and loss decreases the value of r . This approach results in a system less sensitive to changes. In order to adjust the performance measure to the weight of the other interests, we transform the values of n_w as follows:

$$n_w = \frac{(n_w - n_l)}{2} + \frac{n_{total}}{2} = \frac{(n_w - n_l)}{2} + \sum_{j=1}^{n_{total}} \frac{1}{2}$$

Instead of considering the actual number of wins, the effective number of wins $n_{w,i}^{eff}$ w is calculated by adjusting the second term of the expression. This term represents the summation of n_{total} terms.

To take the strength of the other interests into account, these terms are substituted by actual ratings of the interests r_j and the formula for the effective number of wins for an interest i is as follows:

$$n_{w,i}^{eff} = \frac{(n_{w,i} - n_{l,i})}{2} + \sum_{k=1}^{n_{total,i}} X_{ijk} r_j$$

Combining these formulas, the linear equation relating the ratings of an interest and the others is written as:

$$(2 + n_{total,i})r_i - \sum_{j=1}^{n_{total,i}} X_{ijk} r_j = \frac{(n_{w,i} - n_{l,i})}{2}$$

If the total number of interests at an actor is N , we have a linear system with N equations and N variables:

$$C\vec{r} = \vec{b}$$

where \vec{r} , \vec{b} and the Colley matrix C are defined as follows:

$$\vec{r} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_N \end{bmatrix} \quad \vec{b} = \begin{bmatrix} 1 + (n_{w,1} - n_{l,1})/2 \\ 1 + (n_{w,2} - n_{l,2})/2 \\ \vdots \\ 1 + (n_{w,N} - n_{l,N})/2 \end{bmatrix}$$

$$C = [c_{ij}]_{i,j} = 1 \dots n$$

The elements of Colley matrix are defined as follows:

$$c_{ii} = 2 + n_{total,i}$$

$$c_{ij} = -n_{j,i}$$

where $n_{j,i}$ is the number of times the interests i and j were compared to each other.

The matrix C is positive definite [36], which allows the efficient solving of the linear system of C , \vec{r} and \vec{b} using standard techniques. Cholesky decomposition can be used with back-substitution to solve this linear system of equations. The solution of this system would represent the vector of numbers corresponding to the ratings of all N interests, and the resulting rankings are determined by sorting the elements of the solution vector r in decreasing order.

When adapting the Colley method for ranking interests, there are two important constraints: the duration of a period and the minimum change in the sensed values to update weights.

Our system continuously collects information from the environment. The duration of the *time period*, which will be used to compare interests, depends on the particular requirements of the systems. Thus, the elements of Colley matrix become:

$$c_{ii} = 2 + \frac{T_t}{t} \quad c_{ij} = -\frac{T_{j,i}}{t}$$

where T_t is the total time passed, $T_{j,i}$ is the total time that both interest i and interest j are active and t is the chosen *time period*.

The actors update the weights of the interests in their records according to their calculated rankings. However, if the weights are updated at the actor areas for every change in the ranking values of the interests, these updates can create excessive network traffic. Therefore a range (l) is defined to update the network as follows:

$$l = \frac{\sum_{i=0}^N (w_i(\frac{T}{t}) - w_i(\frac{T}{t} - 1))}{N}$$

where w_i is the weight of the interest i .

Obviously, there is a trade-off between the value of l and the number of updates for interests. If we choose a very small value for l , there will be high responsiveness to the changes in observed values with the requirement of frequent updates in each actor area. If l is chosen to be large, then the number of updates decreases. However, if l is very large, the responsiveness to changes will be low and the system will essentially behave as if the weights are predefined.

E. Data transmission

Traditional approaches achieve QoS support in terms of rate guarantees provided to different flows by keeping detailed state information for each flow. The state information includes the expected rate of the flow, the real rate of the flow, update time for the information, and the time window to make decisions. However, this approach is not feasible with low memory and energy resources of sensor nodes. In order to eliminate the per-flow state and high computation requirements, we use a method based on the approach of Stoica et al. [42], which can be described as “having packets carry the state”.

The number of bits in the *rate* field of the packet denotes the rate of the packet R_p . There are state encoding mechanisms in the literature by which large numbers can be represented by small number of bits. Therefore each state variable can be restricted to a predefined number of possible values to minimize the complexity of per packet processing. For instance, eight bits ensure 256 (2^8) different values to assign for the rate of a packet at a sensor node.

When an event is captured by a sensor node, the node checks its subscription table. If there is an active interest for that event, the sensor node generates a data packet to notify the actor. The nodes, which capture events in the areas of interests, start the reporting of the events. The pseudocode of the routing algorithm at each sensor node is presented in Algorithm 2.

Sensor nodes have a predefined maximum transmission capacity called *output capacity* C_o . When an intermediate sensor node receives a data packet in response to a passive interest, it switches this interest’s state to ON. The interests with state ON share the output capacity of the node. We denote the number of these interests with N_f . If it is the first received packet for that interest, the output capacity is reduced by the rate of the packet and saved as the remaining output capacity (C_r) of the node.

1) $C_r > 0$: The received packet doesn’t require any further processing or encoding unless the output capacity is exceeded.

2) $C_r < 0$ and $R_p > R_e$: Packet drops occur when C_r becomes negative. We define the efficient rate (R_e) as the amount of output capacity that the node can fairly employ for a flow when C_r is negative. R_e is formulated as follows:

$$R_e = \frac{C_o}{N_f}$$

If the rate tag on a data packet is greater than R_e , this means the packets of the interest are received with a rate greater than the rate shared for that interest at that node. Hence this interest will be tagged as “greedy” by setting the greedy flag (F_g) of the interest in the subscription table. The number of packets to drop and the method to drop these packets must be determined in order to provide an efficient service to data traffic. It’s important to note that R_e is the maximum value a packet will be encoded with when all flows are received with rates greater than the efficient rate.

There may be packets received with rate values lower than R_e when C_r is negative. In such a case, if all packets are encoded with rate values smaller than or equal to R_e , there will be an excess capacity that is not used. Our algorithm is designed to use this excess capacity since efficient usage of resources is critical in QoS. Furthermore, our main constraint is the rate of data flows and higher rates for flows can be achieved by employing the excess bandwidth. Accordingly, we define the *shared capacity* C_s , the capacity shared among flows that are received with rates greater than R_e . The interests using C_s are defined as *sharing interests*.

The inserted rate value represents the estimate of the flow’s incoming traffic at the next node and it also depends on the importance of the packet. The priorities of interests are specified by their weights. These values are expressed by the sink when registering the interests and more important interests are registered with higher weights. Therefore the new R_p depends on the weight of its interest w_p and the number of sharing interests N_s and it is calculated as:

$$R_p = \frac{C_s \cdot w_p}{\sum_{i=0}^{N_s} w_i}$$

Algorithm 2 Routing in a sensor node

```

1:  $N_f$ : Number of flows on the node
2:  $N_g$ : Number of greedy flows on the node
3:  $C_r$ : Remaining output capacity of the node
4:  $C_s$ : Output capacity used by the sharing flows
5:  $R_e$ : Fair rate for flows on the node
6:  $P_d$ : Dropping probability of a packet
7: if a packet is received by a sensor node then
8:   if the packet is a notification to an interest then
9:     if it is the first packet for that interest then
10:      set  $F_s$  for the interest
11:      increment  $N_f$  by 1, reduce  $C_r$  by  $R_p$ 
12:    end if
13:    if  $C_r > 0$  then
14:      forward the packet
15:    else
16:      if  $R_p > R_e$  then
17:        set the  $F_g$  of the interest
18:        if the  $F_s$  of the interest is unset then
19:          increase  $C_s$  by  $R_p$ 
20:          set the  $F_s$  of the interest
21:        end if
22:        drop the packet with its  $P_d$ 
23:        if the packet is not dropped then
24:          fill the rate field and forward the packet
25:        end if
26:      else
27:        forward the packet
28:        if the  $F_s$  of the interest is set then
29:          reduce  $C_s$  by  $R_p$ 
30:        end if
31:        unset  $F_g$  and  $F_s$  of the interest
32:      end if
33:    end if
34:  else
35:    if the packet is not in response to an interest then
36:      drop the packet (unwanted packet)
37:    end if
38:  end if
39: end if

```

3) $C_r < 0$ and $R_p < R_e$: If R_p of a packet is lower than R_e when C_r is negative, the packet is forwarded without replacing its rate tag. The interest doesn't share C_s with greedy interests, so its flags are unset. A non-greedy interest can become greedy after new packets for different interests are received by the node and since it uses C_s after that instant, its rate value is added to C_s .

In order to insert an exact rate value in the packets, the number of transmitted and dropped packets must be recorded at the sensor node for a period of time, which is not efficient with the limited resources of the sensor nodes. Therefore packet dropping is done probabilistically at sensor nodes using the rate tags, subscription tables and the output capacities. The dropping probability of a packet increases as the difference between the calculated new rate and the rate tag gets larger. The probability to drop a packet is defined as follows:

$$D_p = 1 - \frac{C_s}{N_s \cdot R_p}$$

In contrast to sensors, actors keep state information for each flow. In order to estimate the flow arrival rate, we use the following equation, which depends on the exponential averaging formula in Stoica et al. [42].

$$R_i^{new} = (1 - e^{-T/K}) \frac{l}{T} + e^{-T/K} R_i^{old}$$

where T is the time between the last two packets of the interest i , l is the packet length and K is a constant. The actor nodes insert these flow rates on each packet they transmit. Using an exponential weight $e^{-T_i^k/K}$ provides more reliable estimation for bursty traffic, even when the packet inter-arrival time has significant variance.

Our application scenario is advantageous for using the approach of Stoica et al. [42] compared to traditional internet routing scenarios. In traditional applications, packets are labeled at the edges of the network by using an estimation algorithm and these labels are updated at the core with a probabilistic approach until the packet reaches a boundary of the network. However in our scenario, a packet is injected into the network with the exact rate placed on its label. Then the tag is updated as the packet is transmitted to the closest actor. An actor has information regarding the flows in its interest subscription table. There is no extra need for exponential averaging to estimate the fair share rate at the actor node, which is calculated using the rate and weight values expressed by the sink. By means of the high transmission range of actor nodes, the data is transmitted to the sink via a path formed by actor nodes.

IV. SIMULATION STUDY

A. Simulation environment and metrics

To analyze the performance of the protocol, we carried out a series of simulations in OPNET modeler [43]. All nodes in the system are stationary and in each simulation, a network topology is generated with 60 sensor nodes distributed randomly over the entire area ($200m \times 200m$) and 4 actors at predefined locations. IEEE 802.11 is used as the underlying MAC layer with direct sequence physical characteristics, $8.02 \cdot 10^{-6}$ Watts transmission power, -95 dBm packet reception power threshold and the automatically assigned channel settings of OPNET modeler. The transmission ranges of the sensor and actor nodes are set to 50 and 180 meters, respectively. The data packet size remains constant at 256 bytes.

The simulations can be organized into two main groups. In the first group, we evaluate the performance of the protocol by comparing it to QBRP [34]. The simulation scenarios and the metrics are chosen similar to the ones used in the evaluations of QBRP for the comparison purposes. The metrics are packet loss, control overhead and end-to-end delay. The memory and energy consumption are also introduced as metrics to evaluate the positive effect of our stateless routing approach. The second group of simulations evaluate the performance of the ranking algorithm. These simulations show the effect of ranking algorithm in our protocol for assigning weights to the interests and for the number of required updates in the network.

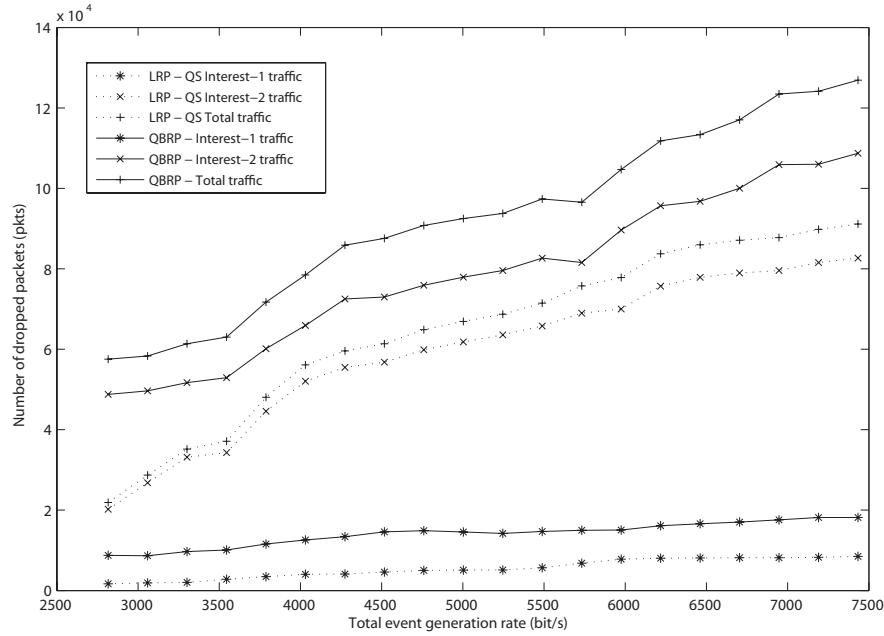


Fig. 3. Packet loss

B. Simulation results

1) *Packet loss:* In order to create data traffic, eight event sources are placed in the field such that each actor area has two event sources, producing events matching Interest-1 and Interest-2. Five times more Interest-2 packets are injected into the network compared to Interest-1. Since QBRP does not define a method for ranking interests, LRP-QS is simulated with constant interest weight values in these experiments.

Figure 3 shows the simulation results for packet loss with increasing event generation rates. The results show the average number of dropped packets with 95% confidence interval for 50 simulation runs at each event generation rate. As expected, for both protocols the number of dropped packets increases with the generation rate of the events. However, the number of packets dropped by our protocol is approximately 60% lower for an event generation rate of 3000 bits/sec. This advantage decreases to about 30% for 7500 bits/sec. Additionally, Interest-1 packets are protected while QBRP is not able to do so. Since both types of the traffic have equal priorities, our algorithm tends to drop Interest-1 packets much less than greedy Interest-2 packets. QBRP drops Interest-1 packets almost twice as much whereas the ratio is much less when we compare dropped Interest-2 packets.

2) *Control overhead:* The number of control packets is critical since increased traffic means more delay and energy consumption. Boukerche et al. [34] showed that number of control packets used by QBRP is not critically affected by the rate of packet generation. Hence we monitor the average number of control packets used by the protocols with non-increasing packet generation rates but varying number of interests. Figure 4 shows that the proposed protocol outperforms QBRP, by using 45% less control packets on the average of 20 simulation runs for each number of interests with 95% confidence interval. Figure 6 shows the energy consumption ratio of the protocols in the same conditions. The energy consumption for data transmission (14.88 mW), reception (12.50 mW) and in the idle mode (12.36 mW) are assigned to the same values as in QBRP for the comparison.

3) *Memory and energy consumption:* The memory consumption is defined in the simulations as the total memory consumed by all nodes. Figure 5 shows the memory consumption ratio of QBRP to LRP-QS with increasing number of interests in the same simulation settings. For each interest number, the sensor nodes are deployed randomly in the area for 50 simulation runs. The positions of the sink and the actor nodes remained the same for each scenario. Figure 5 shows the mean of the ratio and the range for the

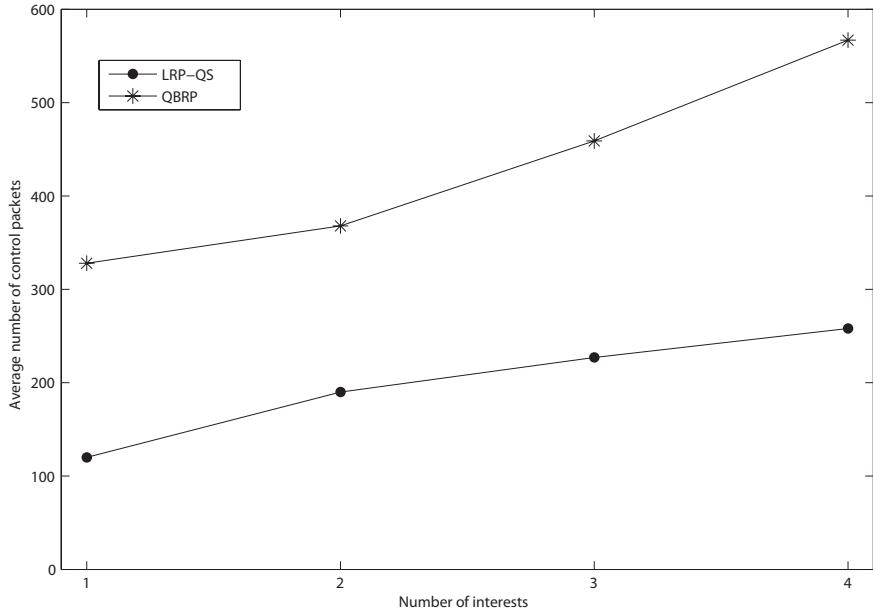


Fig. 4. Control overhead

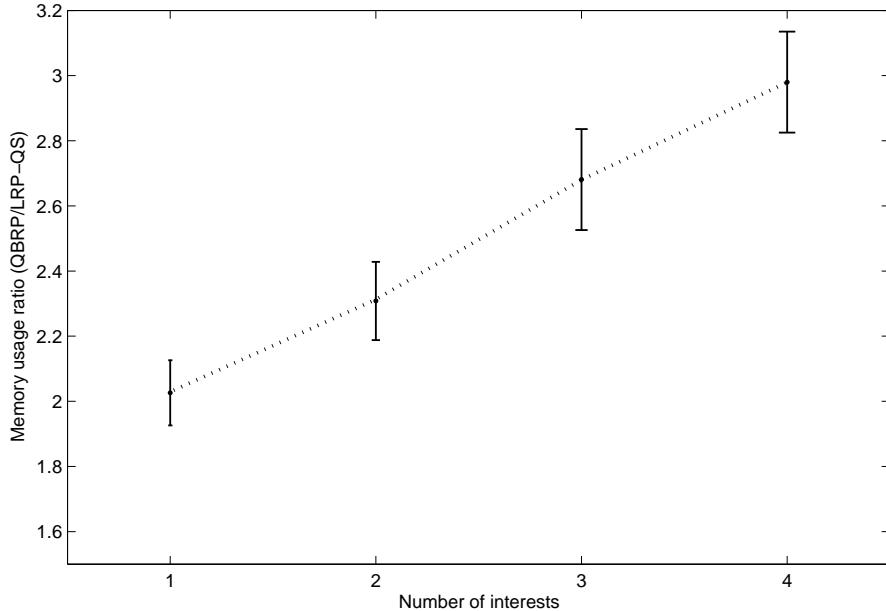


Fig. 5. Memory consumption ratio

ratio values observed in the simulations for each number of interests. We find that our protocol uses less than half of the memory used by QBRP in most cases and performs better with increasing number of interests. QBRP's memory requirements grow with the increasing rate with each additional interest type. As the number of interests exceeds two, the range between minimum and maximum values for memory usage ratio increases. The memory consumption of QBRP depends highly on the locations of interests and the sensor density in these locations. The probability of having different values for these metrics increases with increasing number of interests. Therefore, the ranges for the ratios slightly increase as the number of interests exceeds two. In correlation with the results on control packets, the energy consumption of QBRP is greater than our protocol for all cases and the ratio increases with the increasing number of interests.

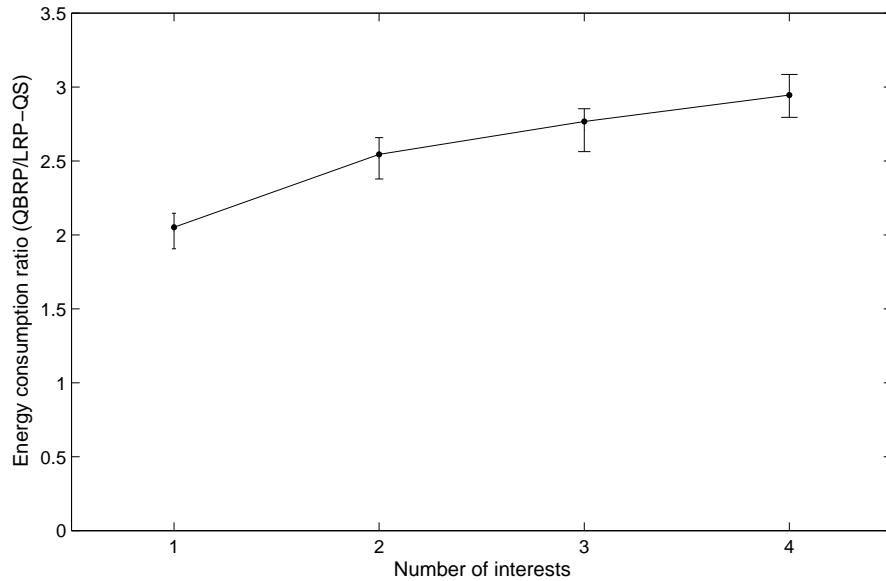


Fig. 6. Energy consumption ratio

4) *End-to-end delay*: Figure 7 shows the end-to-end delay values with increasing number of events for LRP-QS and QBRP in 50 simulation runs for each number of events with 95% confidence interval. QBRP selects paths efficiently with extensive data processing and memory consumption. However, QBRP generates extra traffic by the central processing and the route configuration messages as we have seen in previous simulation results. Our protocol achieves a delay performance similar to QBRP without the increased control overhead, memory consumption or packet drop rate.

Figure 8 presents the average end-to-end delay values with increasing number of sensor nodes for LRP-QS and QBRP with a constant total event generation rate of 200 pkt/s. The protocols have similar delay results for different network sizes and the delay performances of both protocols improve as the number of sensor nodes increase. This result is due to the new communication hops formed by the additional sensor nodes. LRP-QS dynamically adapts to changes and performs better as the new paths are formed. The main sources of delay in QBRP are central processing and route configuration messages, which create extra traffic even when the network size is small. Therefore, the percentage change in the average delay of our protocol is slightly higher than QBRP as the number of sensor nodes increase.

5) *Ranking*: The aim of ranking is to assign weights according to the changes in the values of the collected data at the actor in response to the interests.

In the first set of experiments, the performance of the ranking algorithm is evaluated in an actor area. There are four types of interests, which have equal initial weights. The event sources are placed in the area such that Interest-1, Interest-2 and Interest-3 packets are produced continuously with values in a range of $\pm 50\%$ of their initial values. On the other hand, all events for Interest-4 are produced with the constant initial value.

Figure 9 shows the percentage changes of the collected values of interests compared to the last value observed for the interests. It can be seen that Interest-4 has the highest and Interest-2 the lowest fluctuation.

Table I summarizes the rankings of interests and their percentages in the simulation. Figure 9 and Table I show that Interest-1 and Interest-2 have the highest percentage changes. 58% of the time Interest-1 and 32% of the time Interest-2 is the highest ranked interest. Since Interest-4 has constant values, it always has the lowest ranking.

Figure 10 shows the calculated weight values for each interest during simulation. Interest-1 has the highest ranking as the observed values for Interest-1 has the highest changes. Although Interest-4 has the lowest ranking throughout the simulation, its weight is not reduced to zero in the system. The weight

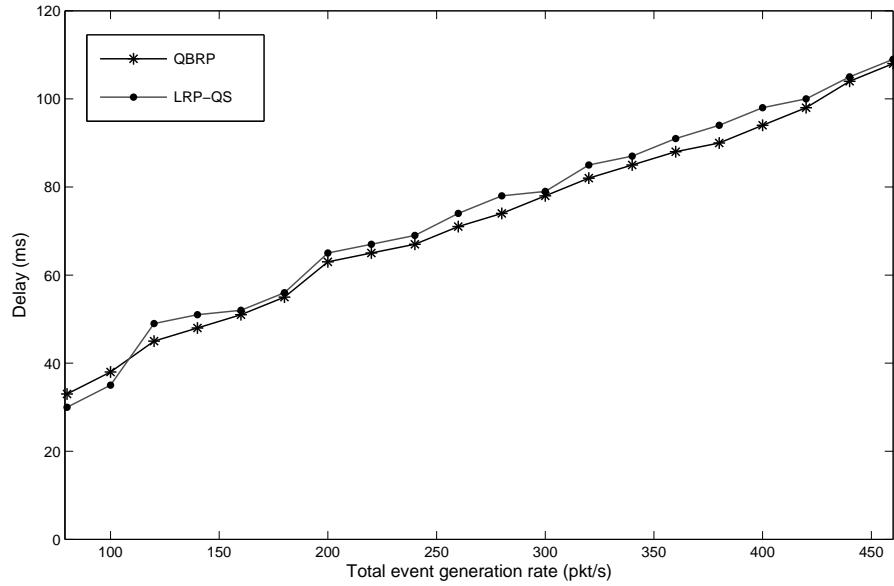


Fig. 7. End-to-end delay

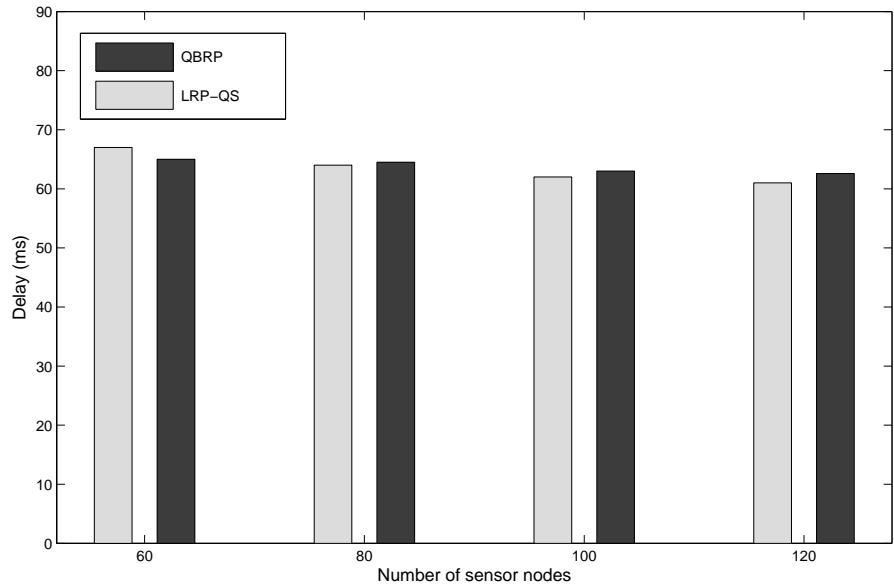


Fig. 8. End-to-end delay for different network sizes

TABLE I
INTEREST RANKS AND PERCENTAGES

Rank	Interest 1	Interest 2	Interest 3	Interest 4
1 st	58%	32%	10%	0%
2 nd	36%	52%	12%	0%
3 rd	6%	16%	78%	0%
4 th	0%	0%	0%	100%

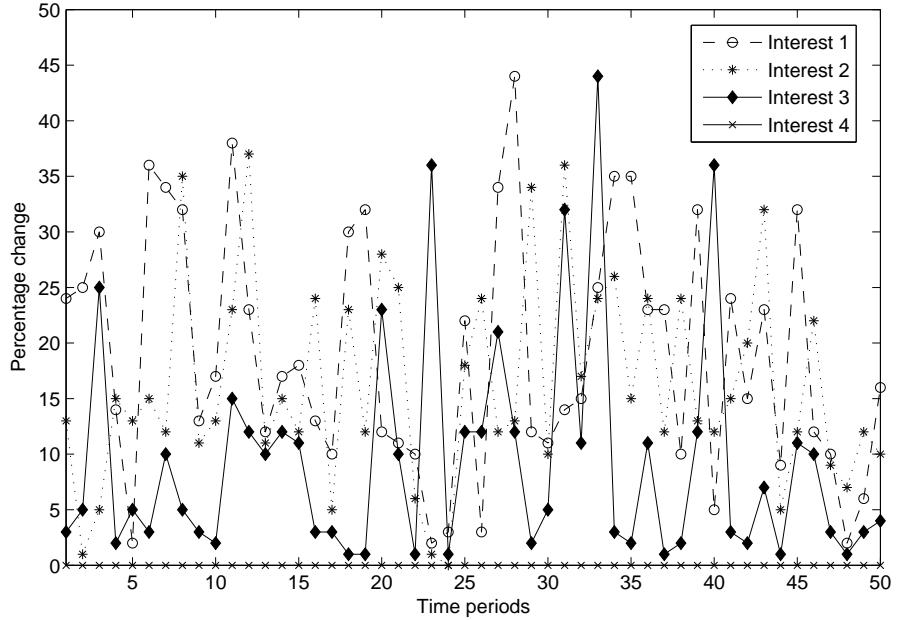


Fig. 9. Percentage changes in the collected values for interests

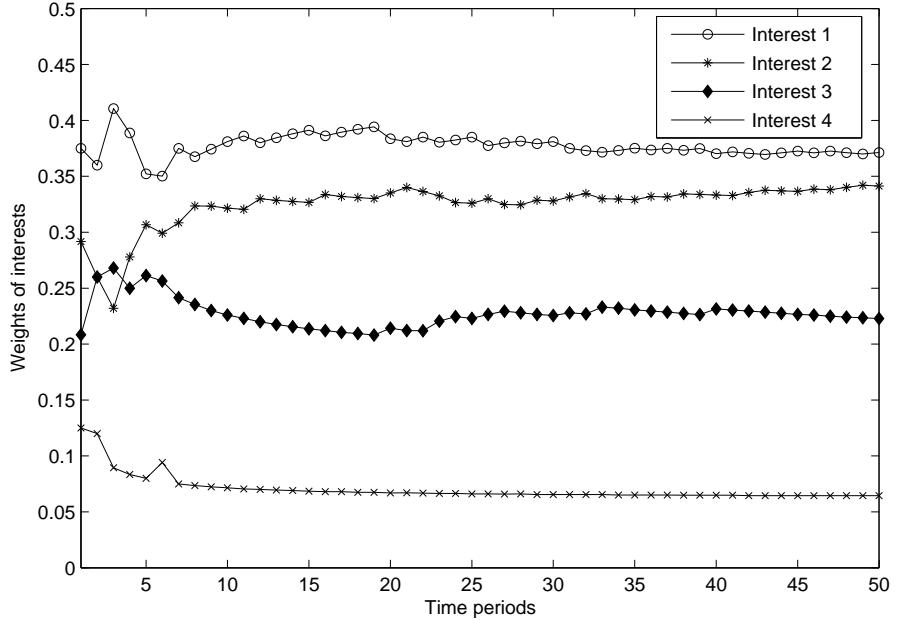


Fig. 10. Weights of the interests

values fluctuate in the first time periods and become more stable after five time periods. Since the set of values is small, the percentage changes in the collected values for interests have a high impact in the beginning of the simulation. The weights of Interest-2 and Interest-3 show a particular behavior because of large spikes of change in their collected data. Additionally, there are spikes in the values observed for Interest-3, which has otherwise small fluctuations. However these spikes do not affect the ranking of the interest drastically, which shows the approach is not sensitive to abrupt changes.

As a comparative analysis, the weight values of each interest are also calculated according to the traditional sports ranking, which depends on the winning percentage. Figure 11 shows that this method results in zero weight for the worst ranked interest in the experiment, which prevents collecting information in response to that interest. Moreover, the highest fluctuations in one time period (wins) are more effective

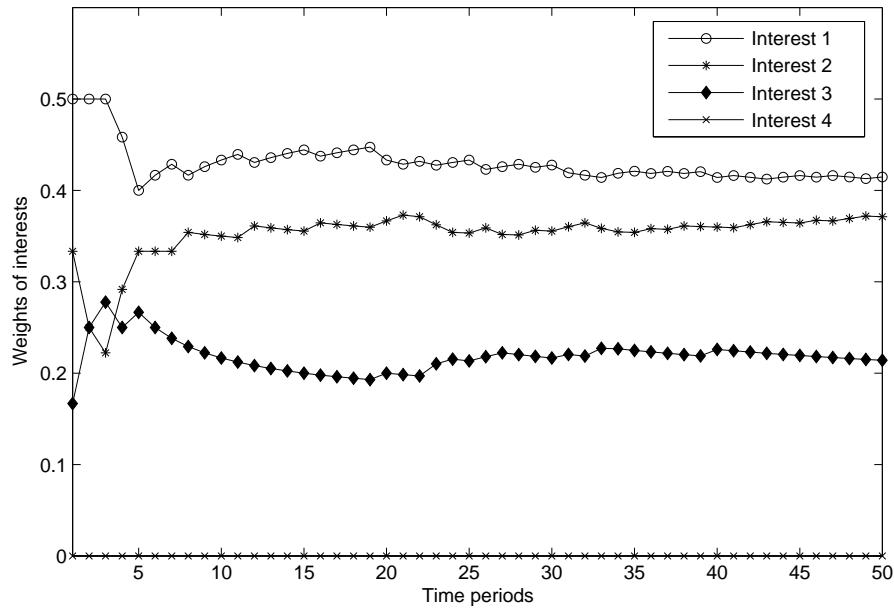


Fig. 11. Weights of the interests with win-ratio ranking

TABLE II
l VALUES AND REQUIRED NUMBER OF UPDATES

<i>l</i> ($\times 10^{-4}$)	288	250	225	200	175	75	63	50	38	25	13
Number of updates	1	2	3	4	5	6	7	8	11	21	33

compared to results in Figure 10, which provides a higher advantage for the winning interest, especially in the beginning of the simulation. While Interest-1 gains approximately a weight of 0.5 with this ranking method throughout the simulation, its average ranking is 0.38 in LRP-QS.

The dynamic ranking improves the performance of the network, however; frequent updates result in increased traffic load and energy loss. Therefore, the defined *l* value for updating the network is critical. Table II shows the required number of updates for different values of *l*. When *l* is set to 8×10^{-3} or higher, five updates are required in the actor's area. In other words, only five updates are required even if the actors are required to update the interests when the sum of all changes in the weights of interests is 3% or more.

In the next set of experiment, the observed values for all interests fluctuate for the first three time periods. Only the collected values for Interest-1 continue this behavior while the rest of the observed values remain stable for the rest of the simulation.

Figure 12 shows the calculated weight values for each interest during simulation. At the end of third time period, Interests 1, 2, 3 and 4 are ranked respectively. The ranking value of Interest-2 is around 75% of the ranking value of Interest-1 whereas the ratio of the ranking values of Interest-4 and Interest-3 is almost 50%. At the end of the third time period, the differences among the weights of interests with stable observed values start to decrease. At the end of the simulation, they acquire similar values while preserving the same ranking order.

The behavior observed in Figure 12 shows that the main deciding factor of ranking interests is the change in the observed values. In addition, Interest-1 is not assigned with an extremely high weight although it is the only interest with changing observed values. Therefore, the system keeps the effect of changes in the beginning of the simulation while giving more importance to fluctuating traffic type.

Figure 13 shows that Interest-1 becomes dominant when traditional win-ratio ranking is used in the same conditions. Interest-2 and Interest-3 lose half of their weights in the 10th time period and the worst

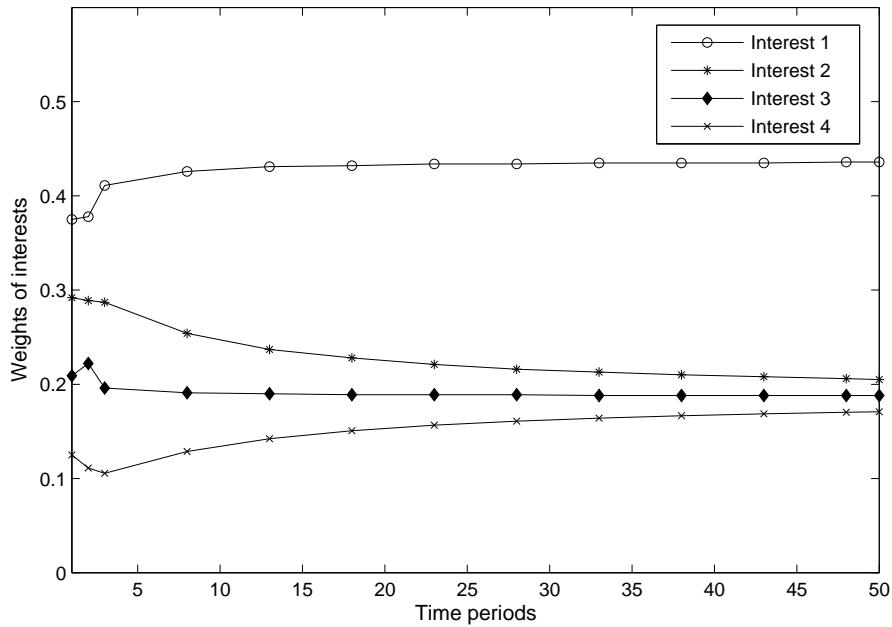


Fig. 12. Weights of the interests

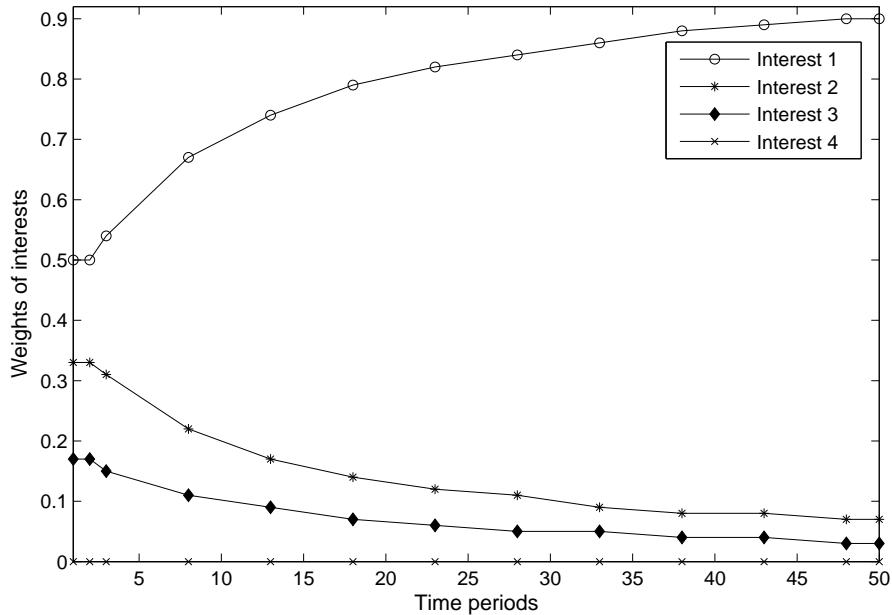


Fig. 13. Weights of the interests with win-ratio rating

ranked interest is assigned with zero weight. Therefore, there is no protection for the traffic of interests other than Interest-1, which wouldn't be acceptable to collect periodical data even if the values do not change.

V. CONCLUSION

In this paper, we introduced a lightweight routing protocol for WSANs. The network is organized so that each actor forms an actor area composed of multiple sensor nodes. Sensor nodes collect information from the environment, the type and importance of which depend on the interests distributed by the sink. The interests have initial weight values, which is dynamically updated according to the changes in the observed values of the events of interest as they occur. Data packets carry rate values on their path to the

actors and they are dropped probabilistically at sensor nodes. The results of the simulation study verify the effectiveness of the proposed scheme compared to QBRP in terms of packet loss, memory consumption and control overhead. We also used a nonsensitive ranking method for dynamic weight assignment to the interests and modified the method according to our requirements. The weights of the interests in LRP-QS are altered according to the changes in the observed values of the events. The simulations show that LRP-QS can dynamically prioritize the interests with a relatively low number of updates in the network. As a part of the future work, we intend to improve our protocol by optimizing the number and locations of deployed actors in the network by taking into account the network area, network size and radio range of both actor and sensor nodes. We also plan to improve the protocol by scheduling node sleep intervals for the sensor nodes.

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, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, Mar. 2002.
- [2] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: research challenges," *Ad Hoc Networks*, vol. 2, no. 4, pp. 351–367, Oct. 2004.
- [3] F. Xia, "QoS Challenges and Opportunities in Wireless Sensor/Actuator Networks," *Sensors*, vol. 8, no. 2, pp. 1099–1110, June 2008.
- [4] S. Basagni, M. Mastrogiovanni, A. Panconesi, and C. Petrioli, "Localized Protocols for Ad Hoc Clustering and Backbone Formation: A Performance Comparison," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 4, pp. 292–306, Apr. 2006.
- [5] 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, Apr. 2002.
- [6] S. Soro and W. B. Heinzelman, "Cluster head election techniques for coverage preservation in wireless sensor networks," *Ad Hoc Networks*, vol. 7, no. 5, pp. 955–972, 2009.
- [7] K. Robinson, D. Turgut, and M. Chatterjee, "Entropy based clustering in mobile ad hoc networks," in *Proceedings of IEEE ICNSC*, April 2006, pp. 1–5.
- [8] D. Turgut, K. Robinson, and M. Chatterjee, "Entropy based clustering in mobile ad hoc networks," *Journal of Ubiquitous Computing and Intelligence (JUCI)*, vol. 1, no. 1, pp. 101–109, April 2007.
- [9] N. Aydin, F. Nait-Abdesselam, V. Pryyma, and D. Turgut, "Overlapping Clusters Algorithm in Ad hoc Networks," in *Proceedings of the IEEE GlobeCom*, Dec. 2010, pp. 1–5.
- [10] G. Smaragdakis, I. Matta, and A. Bestavros, "SEP: A Stable Election Protocol for clustered heterogeneous wireless sensor networks," in *Second International Workshop on Sensor and Actor Network Protocols and Applications (SANPA)*, August 2004.
- [11] N. Aslam, W. Phillips, W. Robertson, and S. Sivakumar, "Extending network life by using mobile actors in cluster-based wireless sensor and actor networks," in *International Federation for Information Processing (IFIP), Wireless Sensor and Actor Networks II*, vol. 264, May 2008, pp. 74–84.
- [12] L. Zhang, Z. Hu, Y. Li, and X. Tang, "Grouping-Based Clustering Routing Protocol in Wireless Sensor Networks," in *International Conference on Wireless Communications, Networking and Mobile Computing (WiCom)*, 2007, pp. 2452–2455.
- [13] W.-P. Chen, J. C. Hou, and L. Sha, "Dynamic clustering for acoustic target tracking in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 258–271, July 2004.
- [14] J. Lin and M. Liao, "A clustering patch hierarchical routing protocol for wireless sensor networks," in *International Conference on Computer Science and Education (ICCSE)*, 2010, pp. 941–948.
- [15] A. Boukerche, A. Martirosyan, and R. Pazzi, "An inter-cluster communication based energy aware and fault tolerant protocol for wireless sensor networks," *Mobile Networks and Applications*, vol. 13, no. 6, pp. 614–626, Dec. 2008.
- [16] W. Hu, N. Bulusu, and S. Jha, "A Communication Paradigm for Hybrid Sensor/Actuator Networks," *International Journal of Wireless Information Networks*, vol. 12, no. 1, pp. 47–59, 2005.
- [17] E. Cañete, J. Chen, M. Díaz, L. Llopis and B. Rubio, "A service-oriented approach to facilitate WSAN application development," *Ad Hoc Networks*, vol. 9, no. 3, pp. 430–452, 2011.
- [18] Y. Tao, Y. Zhang, and Y. Ji, "Flow-balanced routing for multi-hop clustered wireless sensor networks," *Ad Hoc Networks*, vol. 11, no. 1, pp. 541–554, 2013.
- [19] T. Melodia, D. Pompili, V. Gungor, and I. Akyildiz, "Communication and Coordination in Wireless Sensor and Actor Networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 10, pp. 1116–1129, Oct. 2007.
- [20] 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 NEW2AN*, May 2006, pp. 365–383.
- [21] E. Ngai, Y. Zhou, M. R. Lyu, and J. Liu, "A delay-aware reliable event reporting framework for wireless sensor-actuator networks," *Ad Hoc Networks*, vol. 8, no. 7, pp. 694–707, 2010.
- [22] A. Sama and K. Akkaya, "Real-time routing for mobile sensor/actor networks," in *Proceedings of the IEEE LCN*, Oct. 2008, pp. 821–828.

- [23] K. L. Hung, B. Bensaou, J. Zhu, and F. Nait-Abdesselam, "Energy-Aware Fair Routing in Wireless Sensor Networks with Maximum Data Collection," in *Proceedings of the IEEE ICC*, Dec. 2006, pp. 3438–3443.
- [24] 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.
- [25] V. Paruchuri, A. Durresi, and L. Barolli, "Energy Aware Routing Protocol for Heterogeneous Wireless Sensor Networks," in *Proceedings of the Database and Expert Systems Applications*, Aug. 2005, pp. 133–137.
- [26] D. Zhang, G. Li, and K. Zheng, "An energy-balanced routing method based on forward-aware factor for Wireless Sensor Network," *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, pp. 1–1, 2013.
- [27] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed Diffusion for Wireless Sensor Networking," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 2–16, Feb. 2003.
- [28] 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.
- [29] B. C. Villaverde, S. Rea, and D. Pesch, "InRout - A QoS aware route selection algorithm for industrial wireless sensor networks," *Ad Hoc Networks*, vol. 10, no. 3, pp. 458–478, 2012.
- [30] M. Baunach, "Collaborative memory management for reactive sensor/actor systems," in *Proceedings of the IEEE LCN*, Oct. 2010, pp. 953–960.
- [31] O. Banimelhem and S. Khasawneh, "GMCAR: Grid-based multipath with congestion avoidance routing protocol in wireless sensor networks," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1346–1361, 2012.
- [32] J. Ben-Othman and B. Yahya, "Energy efficient and QoS based routing protocol for wireless sensor networks," *J. Parallel Distrib. Comput.*, vol. 70, no. 8, pp. 849–857, Aug. 2010.
- [33] M. Hammoudeh and R. Newman, "Adaptive routing in wireless sensor networks: QoS optimisation for enhanced application performance," *Information Fusion*, no. 0, pp. –, 2013.
- [34] A. Boukerche, R. B. Araujo, and L. Villas, "A novel QoS based routing protocol for wireless actor and sensor networks," in *Proceedings of the IEEE GlobeCom*, Nov. 2007, pp. 4931–4935.
- [35] M. İ. Akbaş and D. Turgut, "Lightweight Routing with QoS Support in Wireless Sensor and Actor Networks," in *Proceedings of the IEEE GlobeCom*, Dec. 2010, pp. 1–5.
- [36] W. N. Colley, "Colley's Bias Free College Football Ranking Method: The Colley Matrix Explained," <http://www.colleyrankings.com/matracte.pdf>, Sept. 2003.
- [37] K. Massey, "Statistical Models Applied to the Rating of Sports Teams, Master's thesis, Bluefield College, Bluefield," Spring 1997.
- [38] L. Ingram, "Ranking NCAA Sports Teams with Linear Algebra, Masters thesis, College of Charleston," Apr. 2007.
- [39] A. Y. Govan, A. N. Langville, and C. D. Meyer, "Offense-defense approach to ranking team sports," *Journal of Quantitative Analysis in Sports*, vol. 5, no. 1, pp. 1–17, 2009.
- [40] T. P. Chartier, E. Kreutzer, A. N. Langville, and K. E. Pedings, "Sensitivity and stability of ranking vectors," *SIAM Journal on Scientific Computing (SISC)*, vol. 33, no. 3, pp. 1077–1102, May 2011.
- [41] T. P. Chartier, E. Kreutzer, A. N. Langville, and K. E. Pedings, "Bracketology: How can math help?" *Mathematics and Sports, Dolciani Mathematical Expositions 43, Joseph Gallian, ed. Mathematical Association of America*, pp. 55–70, 2010.
- [42] I. Stoica, S. Shenker, and H. Zhang, "Core-Stateless Fair Queueing: A Scalable Architecture to Approximate Fair Bandwidth Allocations in High-Speed Networks," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 33–46, Feb. 2003.
- [43] "OPNET Modeler," <http://www.opnet.com>.