Lightweight Routing with QoS Support in Wireless Sensor and Actor Networks

Mustafa İlhan Akbaş and Damla Turgut
School 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) can be used for monitoring physical environments and acting according to the observations. In order to differentiate the actions based on the sensed information, WSANs comprise of various applications with different quality of service (QoS) requirements. QoS solutions for WSANs are challenging compared to traditional networks because of the limited resource capabilities of sensor nodes. In terms of QoS requirements, WSANs also differ from WSNs since actors and sensors have distinct resource constraints.**

In this paper we present LRP-QS, lightweight routing protocol with QoS support for WSANs. Our protocol provides QoS by differentiating the rates among different types of applications with dynamic packet tagging at the sensor nodes and per flow management at the actor nodes. Through extensive simulations we observe a greater packet delivery ratio and a better memory consumption rate in comparison with the related mechanisms.

I. Introduction

Wireless sensor networks (WSNs) [1], which have been studied for more than a decade now, are designed to gather information about their environment. In most of the WSN applications, sensors transmit the sensed data to predetermined nodes or users in the network. However, observation of data is not sufficient in most cases and some kind of a response is also required. An extension of sensor networks that features coexistence of sensors and actors, wireless sensor/actor networks (WSANs) [2] has been emerged. WSANs enable networks and systems not only to sense, but also to act according to the events occurring in the environment.

Sensor nodes in WSANs are generally small devices with very limited data processing capability, transmission rate, energy, and memory. Actor nodes on the other hand have better computation and communication capabilities, increased memory and long lasting batteries. Thus WSANs have a heterogeneous structure in terms of node resources. The availability of resources is critical for quality of service (QoS), so WSANs cannot be simply regarded as WSNs when aiming for QoS support in the network.

In a network with multiple applications, each one has different QoS requirements. Consequently, QoS support becomes a vital part of WSANs, widely used in various applications [3]. For instance, data sensed about an intruder is required to be transmitted with the lowest delay possible, whereas accuracy may be more important for periodically collected information.

In this paper we propose lightweight routing protocol with QoS support (LRP-QS) to increase the lifetime of the network and to provide the best possible rate for each flow. In our scenario, there are bursty data sources resulting in congestion at various points in the network from time to time, which increases the criticality of QoS in the network. In LRP-QS, QoS is defined as the assurance of the services required by the applications and the main optimization objectives are the packet transmission and loss rates. LRP-QS is developed considering the heterogeneous structure of WSANs. At sensor nodes, it is light-weight and efficient in terms of memory and power consumption. Actors have more complicated tasks in accordance with their capabilities, but their resource constraints are also taken into account.

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

The network organization of LRP-QS shares several aspects with cluster-based routing schemes in which the clusterheads have more energy and computation power such as the actors in LRP-QS. The HEED by Younis and Fahmy [4] uses energy as its main parameter to increase the network lifetime and the clusterheads change periodically, which creates a network structure different than LRP-QS. In the technique proposed by Aslam et al. [5], actor nodes find an optimal geographical location with respect to their associated clusterheads. Chen et al. [6] introduced a dynamic clustering algorithm for target tracking, which creates a hierarchical network structure. None of these solutions matches exactly with our requirements such as simplicity and predetermined clusterheads. Our approach in network organization is lightweight for implementation on sensor nodes and makes use of the heterogeneous node structure.

There have been efforts on communication protocols that provide QoS support in WSANs. Melodia et al. [7] designed a sensor-actor coordination model based on an event-driven clustering paradigm, which also includes a trade off between energy consumption and latency. Another coordination algorithm among actors is introduced with the real-time communication framework by Ngai et al [8], where an event reporting algorithm for sensor-actor communication is also given. The routing algorithm by Hung et al. [9] 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. [10] enhances reliability with redundancy. These algorithms focus mostly on energy efficiency of the sensor nodes, but do not study the requirements of the applications. Boukerche et al. [11] proposed QBRP, a routing protocol with service differentiation, in which route generation is done at actors by using the information collected from the sensors. The next hop for each type of packet is sent to the nodes based on the data, which is centrally processed at the actor nodes. Our protocol uses an approach similar to QBRP in network organization, however our approach eliminates heavyweight requirements of QBRP such as central processing and frequent updating.

III. Lightweight Routing with QoS Support

We consider a network comprising of a number of sensor nodes, multiple actors and we use the language popularized by the Directed Diffusion model [12]. That is, we assume the sink expresses a set of interests regarding information to be collected.

QoS is defined, in the context of this paper, as the network capability to utilize the resources efficiently for each packet received by a sensor node. Packets are treated according to the interests in order to meet the sink's requirements. The main criteria when providing QoS are the rate of the flows and employment of the transmission bandwidth.

LRP-QS is composed of four phases described in the following subsections.

A. Phase 1: Formation of actor areas

The sensor nodes have *actor ID* and *hop value* attributes, both of which are null when the network is initialized. In other words, initially a sensor node knows neither the actor it is associated with nor the hop-distance of that actor.

The network configuration starts with the flooding of area configuration packets (ACP) by the actors. An ACP includes the actor's ID and the number of hops the packet is forwarded. 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 only one actor even when it receives ACPs from multiple actors. The node keeps the address of the neighbor node it received its actor's ID as the destination for its data packets. destinations and this record can be leveraged resources. However the main objective is to keep as possible in the sensor nodes.

At the end of phase-1, each sensor node is associated with an actor; it has information about its lower-hop neighbor(s) and the number of hops required to reach its actor.

B. Phase 2: Formation of communication backbone

After actor areas are formed, they need to be linked to transmit the collected data to the sink. We define the network of links among actors and the sink as "communication backbone". The sink starts formation of the communication backbone by sending an area integration packet (AIP) with its ID in the source field. When an actor receives an AIP from the sink, it saves the sink as the destination address for data packets (DA_d). Then the AIP is forwarded in the network among actors. Hence the sink is 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 would receive the collected data through sensor nodes, which would create severe packet loss and delay.

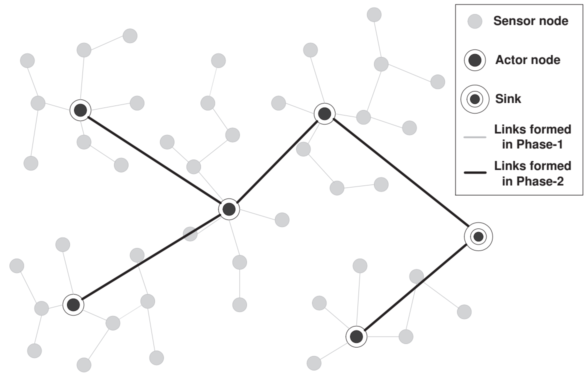

Fig. 1. Hypothetical network view after phase-2

An actor saves the first actor from which it received the AIP as the DA_d and retransmits AIP with its ID on 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. A summary of the pseudocode of the algorithm used at each actor node receiving an AIP is presented in Algorithm 1.

Algorithm 1 Processing of AIP at an actor node

```
Check source address of AIP
Check $DA_d$
if $DA_d$ is a sink then
    drop AIP
else if AIP is received from a sink then
    $DA_d$ = ID of the sink transmitted AIP
    transmit AIP
else if AIP is received from an actor then
    if $DA_d$ is an actor then
        Put ID of the actor that transmitted AIP in $L_r$
    else
        set $DA_d$ = ID of the actor transmitted AIP
        transmit AIP
    end if
else if AIP is received from a sensor node then
    if $DA_d$ is an actor then
        drop AIP
    else
        $DA_d$ = ID of the actor on AIP
        transmit AIP
    end if
end if
```

Sensor nodes use a lightweight algorithm to process AIPs. A sensor node transmits the first AIP from its actor to its higher-hop neighbor(s) and transmits the first AIP from another actor to its lower-hop neighbor to guarantee the conveying of AIPs in the network. All other AIPs are dropped by the sensor nodes. Hence if each actor is guaranteed to be in the transmission range of at least one actor, then the AIPs don't need to be processed at sensor nodes. Figure 1 shows the network view of a LSR-QS network after phase-2.

C. Phase 3: 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 unless the interest is already included in the table. All the information expressed in the interest packet such as type, time, location are stored in this table.

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 is in the area where the sink is interested, then it will generate data packets when it senses an event.

At the end of phase-3, each sensor node knows which interest packets to generate. We define these interests as *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. Phase 4: Data transmission

QoS support in terms of rate guarantees provided to different flows is achieved in traditional approaches by keeping detailed state information for each flow. The state information consists of several parameters such as the expected rate of the flow, the real rate of the flow, update time for the information, and 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. [13], which can be simply described as "having packets carry the state".

The number of bits in *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.

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 in LRP-QS, 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 total number of interests with state "on", (N_f), represents the number of interests sharing the output capacity of this node. 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:

$$R_e = \frac{C_o}{N_f} \tag{1}$$

Algorithm 2 Routing in a sensor node

```
if a packet is received by a sensor node then
    if the packet is a notification to an interest then
        if it is the first packet for that interest then
            set $F_s$ for the interest
            increment $N_f$ by 1, reduce $C_r$ by $R_p$
        end if
        if $C_r > 0$ then
            forward the packet
        else
            if $R_p > R_e$ then
                set the $F_g$ of the interest
                if the $F_s$ of the interest is unset then
                    increase $C_s$ by $R_p$
                    set the $F_s$ of the interest
                end if
                drop the packet with its $P_d$
                if the packet is not dropped then
                    fill the rate field and forward the packet
                end if
            else
                forward the packet
                if the $F_s$ of the interest is set then
                    reduce $C_s$ by $R_p$
                end if
                unset $F_g$ and $F_s$ of the interest
            end if
        end if
    else
        if the packet is not in response to an interest then
            drop the packet (unwanted packet)
        end if
    end if
end if
```

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 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. Accordingly, we define "shared capacity" (C_s), the amount of capacity that is 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 rate of the flow's incoming traffic at the next node and it also depends on the importance of the packet. We use weights to express variety in priorities of interests. 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:

$$R_p = \frac{C_s \cdot w_p}{\sum_{i=0}^{N_s} w_i} \tag{2}$$

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, that 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, number of transmitted and dropped packets must be recorded at the sensor node for a period of time, which is not efficient with 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. 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:

$$D_p = 1 - \frac{C_s}{N_s \cdot R_p} \tag{3}$$

In contrast to sensors, actors keep state information for each flow, so the complexity in data transmission is primarily on actor nodes. In order to estimate the flow arrival rate, we use the following equation, which depends on the exponential averaging formula in Stoica et al. [13].

$$R_i^{new} = (1 - e^{-T/K})\frac{l}{T} + e^{-T/K} R_i^{old} \tag{4}$$

where T is the time between last two packets of the interest *i*, l is the packet length and K is a constant. Actor nodes insert these flow rates on each packet they transmit.

Our application scenario is advantageous for using the approach of Stoica et al. [13] 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. LRP-QS doesn't need an extra exponential averaging to estimate the fair share rate at the actor node, it is calculated by 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 as described in Section III-B.

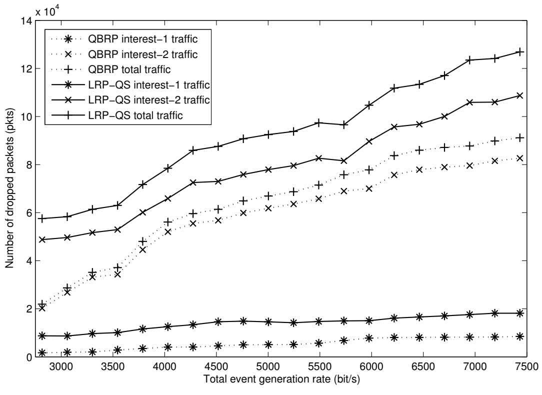

Fig. 2. Packet loss

IV. Simulation Study

A. Simulation environment and metrics

Extensive simulations are carried out in OPNET modeler [14] to analyze the effectiveness of LRP-QS. We assume that all the actor and sensor nodes as well as the sink are stationary. In each simulation, a network topology is generated with the sink located at one side of the area ($200x200m$) and 60 sensor nodes distributed randomly over the entire area with 4 actors at predefined locations. The 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 auto assigned channel settings. The transmission ranges of sensor and actor nodes are taken as 50 and 180 meters, respectively. The data packet size is constant and 256 bytes.

We study the effect of LRP-QS with the simulation metrics: packet loss, control overhead, memory consumption and end-to-end delay. We compare LRP-QS to QBRP [11], so simulation scenarios are chosen similar to the ones used by Boukerche et al. [11]. We also have additional scenarios for evaluating the performance of LRP-QS under certain conditions.

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 interest 1 and 2 events with equal priorities. Interest 2 packets are injected into the network five times more than Interest 1 packets. Figure 2 denotes the packet loss with increasing event generation rates. LRP-QS drops significantly less number of packets than QBRP. As the number of dropped packets decreases, the delivery rate and the reliability of the protocol increase. Additionally, LRP-QS protects Interest 1 packets 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 as LRP-QS whereas the ratio is much less when we compare dropped Interest 2 packets.

2) Control overhead: Number of control packets is critical since increased traffic means more delay and energy consumption. Boukerche et al. [11] showed that number of control packets used by QBRP is not critically affected by the rate

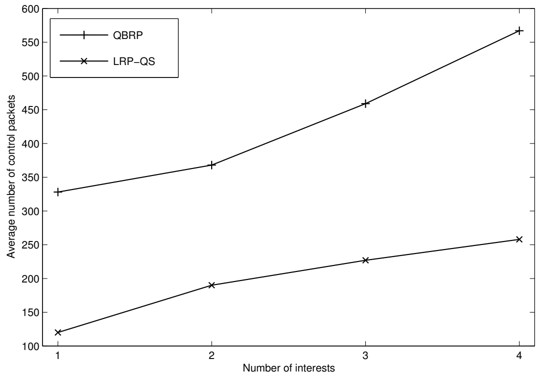

Fig. 3. Control overhead

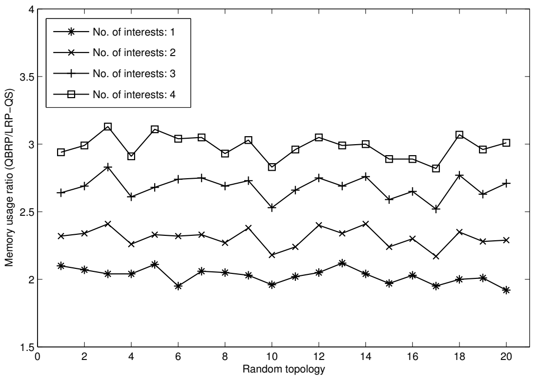

Fig. 4. Memory consumption

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 traffic types in 20 simulation runs. Figure 3 shows that LRP-QS outperforms QBRP, by using 45 percent less control packets on average.

3) Memory consumption: Memory consumption is calculated as the total memory consumed by all of the sensor nodes. Figure 4 denotes memory consumption ratio of the protocols with increasing number of interests in 20 different sensor deployment scenarios. LRP-QS uses less than half of the memory used by QBRP in 95 percent of all cases and LRP-QS also performs better with increasing number of interests. QBRP's memory requirement grows with an increasing rate with each additional interest type.

4) End-to-end delay: Figure 5 shows both protocols have similar end-to-end delay performances with increasing number of events. QBRP selects paths efficiently with extensive data processing and memory consumption. However central processing and the route configuration messages cause QBRP to generate extra traffic as we have seen in previous simulation results. LRP-QS achieves a similar delay performance without the increased control overhead, memory consumption or packet drop rate.

V. Conclusion

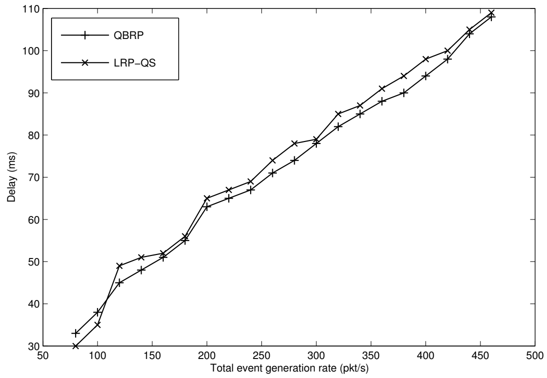

Fig. 5. End-to-end delay

In this paper, we proposed LRP-QS, a lightweight routing protocol that provides QoS for different types of applications in WSANs. The packets carry their rate values on their path to the actors and they are dropped probabilistically at sensor nodes. The proposed protocol is compared to QBRP with extensive simulations and results verify the effectiveness of the proposed scheme compared to QBRP in terms of packet loss, memory consumption and control overhead.

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

[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, October 2004.

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

[4] O. Younis and S. Fahmy, "Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, October 2004.

[5] 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 *IFIP, Wireless Sensor and Actor Networks II*, vol. 264, May 2008, pp. 74–84.

[6] 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.

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

[8] E. C. H. Ngai, M. R. Lyu, and J. Liu, "A real-time communication framework for wireless sensor-actuator networks," in *Proceedings of IEEE Aerospace Conference*, March 2006.

[9] 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'06*, December 2006, pp. 3438–3443.

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

[11] 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*, November 2007, pp. 4931–4935.

[12] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Proceedings of MobiCom'00*, August 2000, pp. 56–67.

[13] 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, February 2003.

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