Routing of High-Priority Packets in Wireless Sensor Networks

Erol Gelenbe[a], Edith Ngai[b] and Poonam Yadav[c]

[a,c]Intelligent Systems and Networks Group, Imperial College, London SW7 2BT, UK

[b]Department of Information Technology, Uppsala University, Sweden

ABSTRACT

The Random Re-Routing (RRR) algorithm has recently been introduced to provide fast adaptive priority routing to rapidly convey packets of important events in sensor networks, while forwarding routine and lower priority traffic along some secondary network

paths. This technique can be applied for sensor networks which monitor the environment by a large amount of sensors distributed in different locations. Such a network has to report large volumes of slowly varying routine data and must quickly report the rarer but more significant events that require immediate attention. In this paper, the RRR algorithm and its implementation in real sensors are presented. Experimental results are provided to show the performance of the algorithm in our sensor network testbed.

Keywords: Wireless Sensor Network, Routing, Significant Events, Experiments, Performance

1. ROUTING IN WIRELESS SENSOR NETWORKS

Wireless sensor networks (WSNs) are deployed for both military applications and surveillance, as well as in many civilian application areas.[1, 2] They are expected to provide robust and reliable services over long period of time and be able to operate with minimal monitoring. WSNs typically forward significant data promptly and efficiently from a number of distinct and geographically distributed sensors to the sinks. Many WSNs have to deal with routine sensing which results in a steady and continuous volume of data, while unusual events of particular interest may occur unexpectedly in which the related information will require fast transmission to the sink(s). Thus packets containing information of the unusual events need “better QoS” in the network, such as shorter end-to-end delay, lower packet loss rate, possibly higher bandwidth and higher security level.

The Random Re-Routing algorithm which is the focus of this paper, is designed to offer a real-time adaptive capability to detect unusual events and provide them with significantly better QoS in the presence of larger volumes of routine data traffic.[3–5] Both analysis based on diffusion approximations[6–8] and simulation have been provided to show its strong potential to satisfy the specific needs for differentiated QoS in WSNs. However the performance of RRR in real WSNs has not yet been investigated. In this paper, we describe the implementation of RRR in a WSN testbed based on Motes and report the experimental results. We evaluate the performance of RRR in terms of packet travel delay in a network with varying traffic loads. Since the routing policy in RRR is controlled by a traffic threshold, we also evaluate its influence to the overall network performance.

The remainder of the paper is organized as follows. Section 2 describes the related work. Section 3 presents the Random Re-Routing algorithm and its implementation in real sensors. Section 4 summarizes the experimental results. Finally, conclusions are given in Section 5.

Further author information: (Send correspondence to E.G.)
E.G.: E-mail: e.gelenbe@imperial.ac.uk, Telephone: +44(0)2075946342
E.N.: E-mail: edith.ngai@it.uu.se
P.Y.: poonam.yadav07@imperial.ac.uk

2. RELATED WORK

Multihop routing algorithms have been studied to meet the requirements of various applications in sensor networks considering their robustness, stability, fairness and optimality. Packet travel delay is always a concern among different performance metrics. Other parameters are also proposed to measure the link quality which could be a function of distance, bandwidth, average traffic, communication cost, mean queue length, measured delay and router processing speed, etc.[9]

A number of real-time communication protocols for sensor networks have been proposed. For instance, He et al.[10] propose SPEED, a protocol which combines feedback control and non-deterministic quality of service (QoS) aware geographic forwarding, while Lu et al.[11] describe a packet scheduling policy, called Velocity Monotonic Scheduling, which inherently accounts for both time and distance constraints. Felemban et al.[12] propose Multi-path and Multi-Speed Routing Protocol (MMSPEED) for probabilistic QoS guarantee in WSNs. Multiple QoS levels are provided in the timeliness domain by using different delivery speeds, while various requirements are supported by probabilistic multipath forwarding in the reliability domain; our approach has some similarity to this work. Huang et al.[13] consider a spatiotemporal multicast protocol, called “mobicast”, which provides reliable and just-in-time message delivery to mobile delivery zones. Some routing protocols with congestion awareness have also been proposed for ad hoc networks.[14, 15]

In this work, we focus on the quality of service in forwarding routine data and unusual events in WSNs, so that the network can offer uncongested paths adaptively for delivering significant traffic emanating from unexpected events, in addition to routing the routine parts of the traffic. Particularly, we provide a practical implementation of an adaptive routing algorithm in real sensors and conduct experiments in our sensor network testbed. The work is different from most existing studies on adaptive sensor network routing based on simulations.[4, 5]

3. THE RANDOM RE-ROUTING (RRR) ALGORITHM

Random Re-Routing (RRR) is a distributed and adaptive routing algorithm which can detect the occurrence of unusual events and provide better QoS for packets that carry information of these unusual events. The WSN is composed of sensor nodes which may have both sensing and forwarding roles, while some of them may be specialized to a forwarding role. RRR is implemented in each of the sensor and forwarding nodes. It distinguishes packets of routine data and unusual events; packets from unusual events are routed along preferred paths, while the routine data are randomly shunted to slower and possibly longer secondary paths.[4, 5]

In RRR, the sensor and forwarding nodes will change their routing policy adaptively according to the current traffic level. When the overall total traffic level is low, the preferred path will be shared by *all* packets. However, when the total traffic exceeds a given threshold, the preferred paths will be reserved for forwarding only the unusual event data packets and secondary paths will be used for the routine data packets. Although this mechanism can in fact increase the total average traffic in the network due to the increased path length of routine data, it will provide significantly better QoS to unusual events. We also need to emphasize that RRR does not rely on any centralized control. All decisions are taken in a distributed fashion at the sensor and forwarding nodes.

The main steps of the RRR algorithm are shown in Figure 1 and described as follows:

1. Each source node will classify a packet either as normal (routine) or priority (unusual), based on whether the data in the packet differs significantly or not from the content of the previous packets from the same source. If the packet's content is indeed significantly different from the previous packets, then the priority bit in the packet header will be set to one (unusual). The packet will be routed according to the decision of the route selection function.

2. The route selection function operates with a high priority traffic measurement window. If a source or transit node measures an incoming packet rate of the priority (unusual) packets that is above a given traffic threshold θ_U, then it chooses to forward all higher priority packets to the destination along the

best QoS path. All lower priority packets will be directed along randomized routes which spread the lower priority traffic across the network away from the high priority paths, reserving better paths to high priority traffic. If the arrival rate of higher priority packets drops below θ_U, the nodes will revert and route all the packets along the better QoS or shortest paths to their destination.

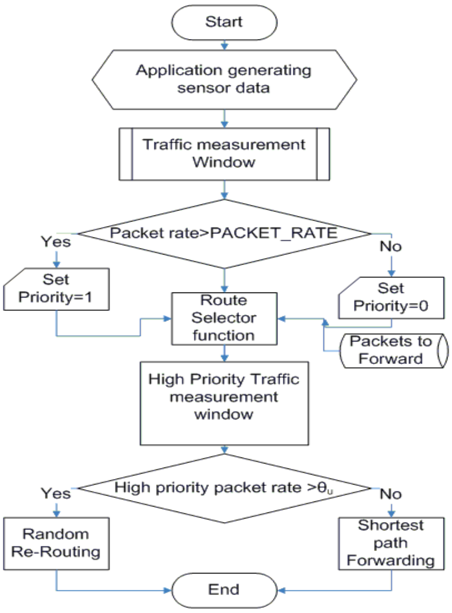

Figure 1. Random Re-Routing algorithm Flowchart.

3.1 Implementation

The implementation of RRR multihop routing algorithm follows the architecture of standard TinyOS 1.x MultihopRouter.[16] The Motes route packets to the destination based on the number of hops to the sink, instead of their physical distances.

The Multihop routing message, the route discovery beacon message and the acknowledgement message are encapsulated in the TinyOS standard message TOS_Msg. The whole implementation is divided into four modules, including route discovery, neighbor table update, assignment of priority to packets and route selection based on the priority of packets.

In the route discovery phase, each nodes broadcasts the route discovery beacon message, which contains its parent node address and its minimum hop count to the sink. Each node which receives the beacon message will compare the received hop count with its own hop count. If the new hop count is smaller, then the node will change its parent to the beacon sender and update the hop count. Each node maintains a neighbour table which is based on hop count in our implementation. A node uses the beacon messages and updates the neighbour table in a round-robin fashion with an increasing order. In our implementation, the route discovery function is disabled. Instead, we use a pre-defined static neighbour table which is built according to a specific network topology.

The measurements were taken in a sensor network testbed composed of 34 Telosb motes randomly placed in our laboratory. For measurement and development purposes, each mote is connected to a server over the Ethernet. We have programmed the motes into three types of nodes: N_E unusual data generating nodes, N_R routine data generating nodes, and a receiver (sink) node. We have considered two network topologies, one with asymmetric links as shown in Figure 2 and the another with the same topology but symmetric links. In the topology with asymmetric links, node ID(251) is the sink node and node ID(102) is the reference node.

Since the space of the laboratory is limited, each node can almost receive packets from every other node. However, in order to test a routing algorithm such as RRR, we need to create a multi-hop network. We achieve this by hard-coding the network topology described earlier and tuning the communication energy level of the nodes to minimum. A node will only receive packets from its neighbours according to in our topology. In our experiments, all nodes report routine data to the sink constantly, while every sensor has a probability p_u to be the source of an usual event, i.e. $p_u = N_E/N$, where $N = N_R + N_E$, to generate unusual event data independent from the others. This independence assumption may be unreasonable when uncorrelated events are being reported across a sensory field.

Under normal conditions the sensors report routine data to the sink at a low data rate, while unusual events may occur infrequently. Our experiments are run with different number of unusual event sources that generate packets at a higher traffic rate. We also introduce an artificial probability f which indicates the packet loss rate at each node.

The node positions are distributed randomly within a 10×10 square meter area, while the sink is located at the center of the square. The motes' clocks are not synchronized as time synchronization may not be available in every network. The measurements of packet delay are carried out by one of two approaches that we have implemented and tested. In the first approach all packets sent out by the senders and received by the sink are delivered by USB wires and logged at the central system. Time stamping is performed by the central system clock and packet delays are then calculated. In the second approach an acknowledgment is returned to the sender, so it can estimate the forward delay by the average round-trip (forwarding the packet and receiving the acknowledgement), assuming that the sender's clock skew is zero. The experimental results reported below adopt the second approach. However, the end-to-end acknowledgements are sent only to a limited number of selected nodes to avoid extra overhead in the network.

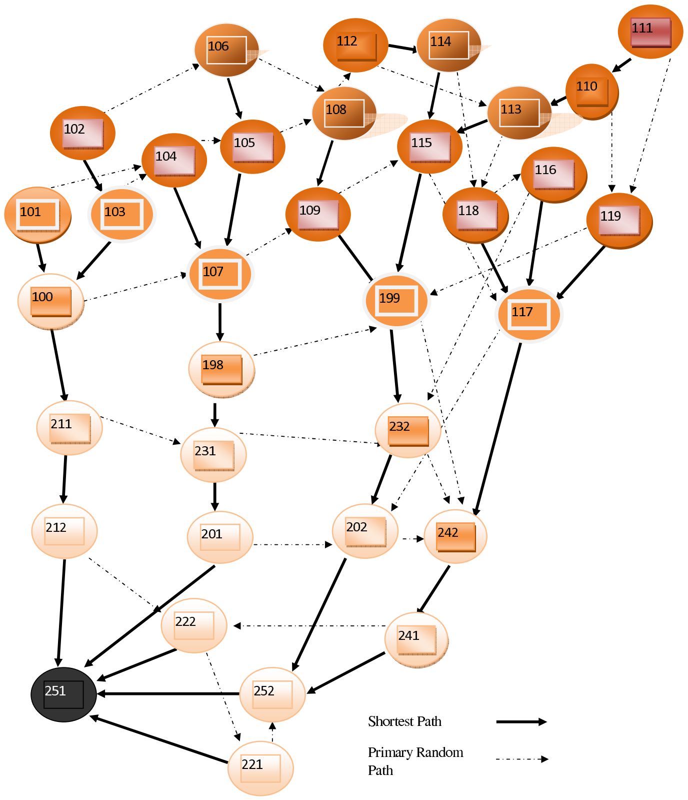

Figure 2. Network topology with asymmetric links.

4. EXPERIMENTAL RESULTS

We conduct extensive experiments to evaluate the packet travel delay with the RRR algorithm by varying N_E, N_R, and θ_U. The parameters chosen are summarized in Table 1, and have been selected so as to be comparable to those of other studies.[10, 12, 17] We have fixed the routine data packet generation rate at 0.2 pkt/sec and the unusual event traffic rate at 0.5 pkt/sec. The round trip packet travel delays are plotted with different number of unusual event sources $N_E = 3$, 5 and 7.

Table 1. RRR Experiment Parameters

Network area	10m*10m
Number of sensor	34
Sensor distribution	Uniform random (random clustered)
Location of Sink	Center of area
Radio range	Maximum
MAC layer	IEEE 802.15.4
Unusual event sources	2 or 3 or 5 or 7 or 11
Probability that a source node sends routine data	p
Packet loss probability	f
Time-out constant ξ	$1/r$
Delay for retransmission M	0.02s
Data rate of unusual events	λ_U
Data rate of routine data	λ_R

In the first experiment, we set the initial distance $H = 5$ in number hops from the source to the sink. A few nodes are selected as unusual event sources denoted by N_E. All remaining nodes are generating routine data. We adopt the Shortest Path Forwarding (SPF) algorithm to deliver all the packets. As expected, Figure 3 shows that the packet delay increases when the the number of unusual event sources N_E increases.

Next, we show how the traffic threshold θ_U works in triggering the RRR algorithm and we compare the performance of RRR with SPF. Figure 4 shows the packet travel delay with $N_E = 3$ and $\theta_U = 0.9$ pkts/sec. The packet delay of unusual events in RRR is comparable to that of all packets in SPF. Since the traffic load is low, the network can handle both unusual events and routine data by the shortest paths without causing much congestion.

Figure 5 shows the packet travel delay with $N_E = 11$ and again $\theta_U = 0.9$ pkts/sec. The network load becomes higher now as there are more sources of unusual events, so the unusual event traffic rates at the intermediate nodes may exceed θ_U. RRR is then activated and the unusual event packets are forwarded along the shortest paths whereas the routine data packets are forwarded via random alternative paths. The unusual event data can achieve lower packet delay than the routine data in RRR. The packet delays of both routine data and unusual events in RRR are lower than the packet delay in SPF. Therefore, RRR algorithm works effectively in this case.

Finally, the effect of the traffic threshold is shown in Figure 6. When the threshold becomes too high, the resulting delay is very large for both the routine and high priority traffic. The reason is that these two classes of traffic share the same shortest paths resulting congestion in the same area. When the network adopts lower threshold value, RRR is activated more often. The two classes of traffic then use different paths and the overall congestion is reduced. Hence, the delay of unusual events in RRR is reduced. Intuitively, the threshold could be chosen by considering the traffic rate λ_R and the network size.

5. CONCLUSIONS

This paper has described a real implementation of the Randomized Re-Routing (RRR) algorithm in a sensor network testbed which provides better QoS for unusual event data in the presence of routine data flows. We

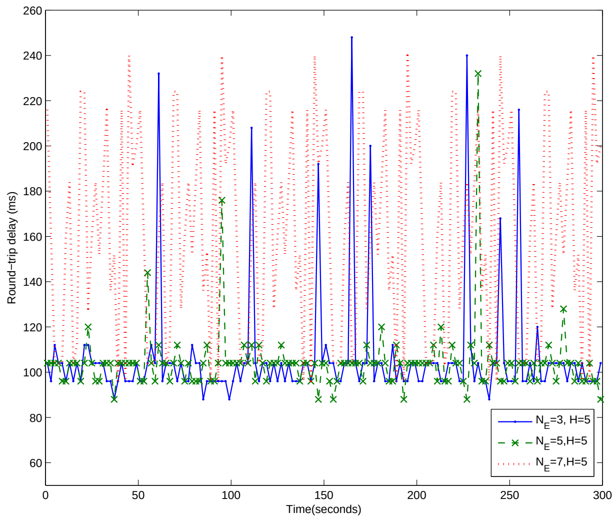

Figure 3. Round-trip packet travel delay with $f = 0.1$, $M = 0.02$, $p = 1.0$, $\lambda_U = 0.5$pkts/s, λ_R =0.2pkt/s and $H = 5$ varying the number of unusual event sources N_E.

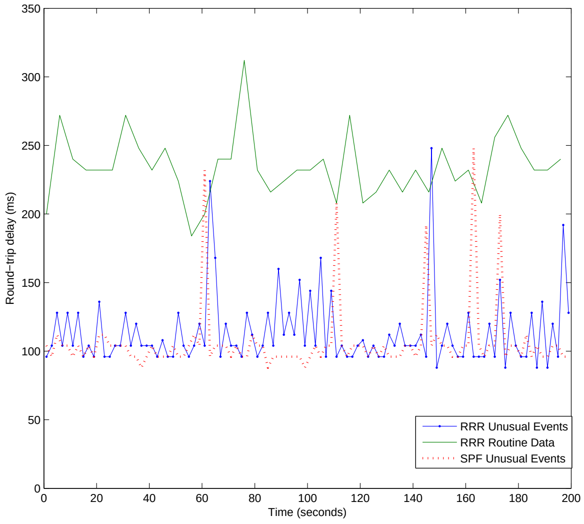

Figure 4. Comparison of round-trip packet travel delay between RRR and SPF with $f = 0.1$, $M = 0.02$, $\lambda_U = 0.5$pkts/s, $\lambda_R = 0.2$pkt/s, $\theta_U = 0.9$pkts/s and $N_E = 3$.

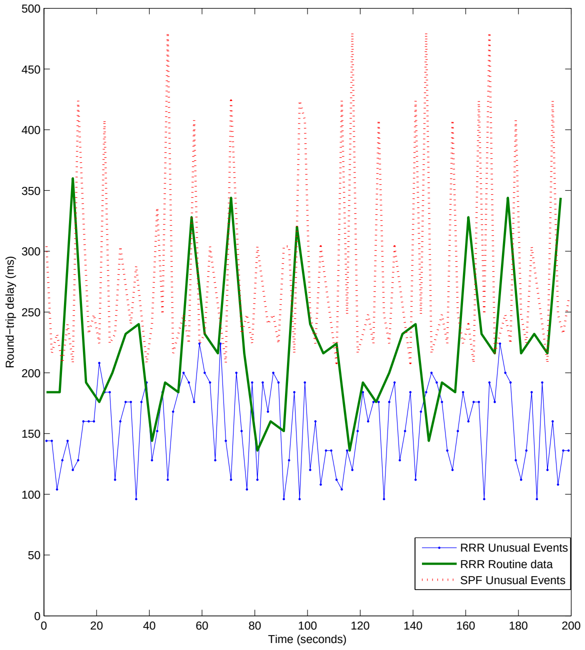

Figure 5. Comparison of round-trip packet travel delay between RRR and SPF with $f = 0.1$, $M = 0.02$, $\lambda_U = 0.5$pkts/s, $\lambda_R = 0.2$pkt/s, $\theta_U = 0.9$pkts/s and $N_E = 11$.

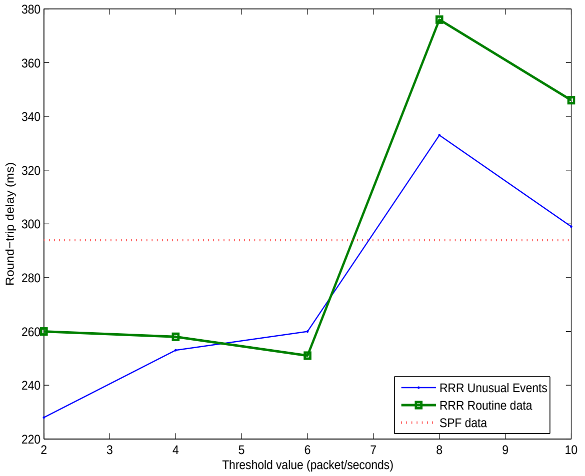

Figure 6. The effect of traffic threshold: Comparison of average round-trip packet travel delay between RRR and SPF against $\theta_U = 2,\ 4,\ 6,\ 8,\ 10$ with $\lambda_U = 2$pkts/s, $\lambda_R =$1pkt/s, $p = 1.0$ and $p_u = 0.27$.

have discussed the experimental setup and evaluated the performance of the RRR algorithm in a network with various traffic loads. We have also showed how the traffic threshold value affects the network performance. The experimental results confirmed that RRR can provide better QoS for unusual events, while offering satisfactory QoS to routine data. In the future, we will further study the selection of the best threshold value for activating random re-routing and study the influence of the MAC layer protocols and wireless interference to routing and the overall network performance.

Acknowledgement

This research was sponsored by US Army Research laboratory and the UK Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the U.S. Government, the UK Ministry of Defense, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES

[1] K. Romer and F. Mattern, “The design space of wireless sensor networks,” *IEEE Wireless Communicationss*, vol. 11, no. 6, pp. 54–61, Dec. 2004.

[2] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, “Wireless sensor networks for habitat monitoring,” in *WSNA ’02: Proc. of the 1st ACM international workshop on Wireless sensor networks and applications*. New York, NY, USA: ACM, 2002, pp. 88–97.

[3] E. Gelenbe and E. C.-H. Ngai, “Random re-routing in sensor networks,” in *Proc. of Annual Conference of International Technology Alliance (ACITA)*, Sep 2008.

[4] ——, “Adaptive qos routing for significant events in wireless sensor networks,” in *Proc. of IEEE MASS*, Atlanta, Georgia, USA, Oct 2008.

[5] ——, “Adaptive random re-routing for differentiated QoS in sensor networks,” in *Proc. of BCS International Academic Conference*, 2008.

[6] E. Gelenbe, “On approximate computer system models,” *Journal ACM*, vol. 22, no. 2, pp. 261–269, Apr 1975.

[7] ——, “Travel delay in a large wireless ad hoc network,” in *Proc. of 2nd Workshop on Spatial Stochastic Modeling of Wireless Networks*, Boston, April 2006.

[8] ——, “A diffusion model for packet travel time in a random mult-hop medium,” *ACM Trans. on Sensor Networks*, vol. 2, no. 3, June 2007.

[9] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, “A high-throughput path metric for multi-hop wireless routing,” *Wirel. Netw.*, vol. 11, pp. 419–434, 2005.

[10] T. He, J. Stankovic, C. Lu, and T. Abdelzaher, “SPEED: a real-time routing protocol for sensor networks,” in *Proc. of IEEE ICDCS*, Providence, RI, U.S., May 2003, pp. 46–55.

[11] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, and T. He, “RAP: a real-time communication architecture for large-scale wireless sensor networks,” in *Proc. of IEEE RTAS*, San Jose, CA, U.S., Sep 2002.

[12] E. Felemban, C.-G. Lee, and E. Ekici, “MMSPEED: multipath multi-speed protocol for QoS guarantee of reliability and timeliness in wireless sensor networks,” *IEEE Trans. on Mobile Computing*, vol. 5, no. 6, pp. 738–754, Jun 2006.

[13] M. Caccamo, L. Y. Zhang, L. Sha, and G. Buttazzo, “An implicit prioritized access protocol for wireless sensor networks,” in *Proc. of IEEE RTSS*, Austin, TX, U.S., Dec 2002, pp. 39–48.

[14] D. Tran and H. Raghavendra, “Routing with congestion awareness and adaptivity in mobile ad hoc networks,” in *Proc. of IEEE WCNC*, Mar 2005.

[15] W. H. Lee, J. M. Chang, and Y. Hasan, “Dynamic memory measuring tool for C++ programs,” in *Proc. of the 3rd IEEE Symposium on Application-Specific Systems and Software Engineering Technology (ASSET 2000)*, Richardson, TX, 2000.

[16] P. Levis, "Ad-hoc routing component architecture," *In the TinyOS documentation site, available in http://www.tinyos.net/tinyos-1.x/doc/ad-hoc.pdf*, February 2003.

[17] E. C.-H. Ngai, Y. Zhou, M. R. Lyu, and J. Liu, "Reliable reporting of delay-sensitive events in wireless sensor-actuator networks," in *Proc. of IEEE MASS*, Vancouver, Canada, Oct 2006.