Intermittent Opportunistic Routing Components for the INET Framework *

Edward Longman, Mohammed El-Hajjar, Geoff V. Merrett

Electronics and Computer Science, University of Southampton, United Kingdom
{el7g15,meh,gvm}@ecs.soton.ac.uk

Abstract

Intermittently-powered wireless sensor networks (WSNs) use energy harvesting and small energy storage to remove the need for battery replacement and to extend the operational lifetime. However, an intermittently-powered forwarder regularly turns on or off, which requires alternative networking solutions. Opportunistic routing (OR) is a potential cross-layer solution for this novel application, but due to the interaction with the energy storage, the operation of these protocols is highly dynamic. To compare protocols and components in like-for-like scenarios we propose module interfaces for MAC, routing and discovery protocols, that enable clear separation of concerns and good interchangeability. We also suggest some candidates for each of the protocols based on our own implementation and research.

1 Introduction

Wireless sensor networks (WSNs) are typically powered by batteries, which can be topped up by energy harvesting, but once empty they cannot be restarted. In future Internet of Things (IoT) network solutions that do not require batteries but still with long lifetimes are required. In intermittently-powered networks, nodes use energy harvesting and a small energy store [1], but to reduce footprint and cost, the device is powered sporadically, as shown in Figure 1. However, current communications techniques for such devices rely on high capability forwarding nodes being within range, or being visited by mobile nodes [1, 2]. Instead, we focus on multi-hop communication between intermittently-powered devices [3, 4], where communication events are limited by the energy storage and nodes are inherently only intermittently-connected.

Opportunistic routing (OR) is a viable networking paradigm for intermittently-powered networks [1], as in Figure 1, that uses flexible forwarding to reach a specified network destination. Flexible forwarding is extremely important in intermittently-connected networks, where a complete route, from the source to the destination, cannot be established. Finding which OR method to use or improve depends on factors such as the node range, inter-node contact time, node energy supply and anticipated network throughput.

2 Networking of Intermittently Powered Devices

Intermittently-powered devices frequently and unpredictably shutdown, and then restart when there is sufficient stored energy. This dynamically affects the operation of the network because devices experience varying loads, interleaved duty cycle [3] or shut down whilst waiting to forward data. Consequently there is a need for simulation with the power consumption in the loop, and across multiple nodes.

*This work was partially supported by the UK EPSRC under EP/P010164/1.
The source code is available at: https://github.com/UoS-EEC/INET-opportunistic-routing

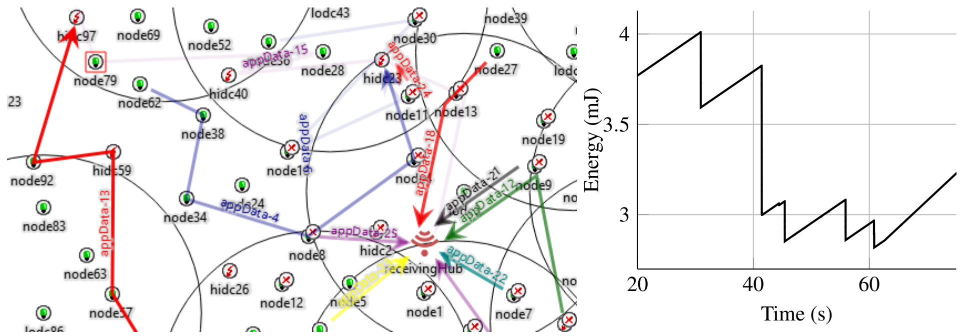

Figure 1: Left: OR multi-point routing showing route taken through intermittently-powered nodes, where nodes marked with cross are off. Range marked by circles around some nodes. Right: Energy profile of harvest-transmit-cycle, where communication causes shutdowns to replenish the energy.

Existing OR implementations for OMNeT do not consider power consumption [2] and are instead designed for higher capability mobile device scenarios. However, WSN nodes have limited energy available to communicate position, availability and routing sets. Additionally, data copying, bundle and broadcast methods exhaust network energy supplies, since data is duplicated and transmitted more than necessary.

Therefore, we propose OR implementations for INET that harness the existing energy storage, traffic and advanced radio models. Since most opportunistic and cooperative routing protocols share common features and requirements this motivates our proposed interface.

3 Common Opportunistic Behaviour and Requirements

OR operates across the DLL and NET layers, harnessing a broadcast link to reach several potential forwarding nodes. When the routing layer has data to send, it is passed down to the MAC layer with attached forwarder selection criteria to select permitted forwarding nodes. When the packet is broadcast, *receiving nodes* must decide whether they meet the criteria for forwarding, and attempt reception of the packet using a cross layer query, as shown in Figure 2. Following acceptance, there may be further MAC layer contention for the packet before it is sent to the routing layer.

The first proposal is the interface, `IOpportunisticLinkLayer`, that should be implemented by any MAC protocol that will check the acceptance criteria using a hook or message before contending to become the forwarder of a received packet. Any response from the routing layer is timing dependent and must therefore occur before the receive window elapses. This is shown in Figure 2, by the direct connection between the Routing Table and MAC. The corresponding interface is implemented by the routing table, `IForwardingJudge`.

Regardless of the OR mechanism, the routing layer must build up a model of the network for direct addressing, a preferred forwarder list and calculating a progress metric. It also must store and disseminate information about the reachable set of neighbors.

The second proposal is a signal-subscribe interface, `ILinkOverhearingSource`, that emits signals for each type of encounter, for example whether it is coincidental or expected in response to an acknowledgment. The receiver of these emitted signals must be able to understand MAC layer datagrams and should then implement `IRoutingOverhearingSource`. This module could predict neighbor availability and implement `IDeferTransmission` to improve reception probability [4].

Following from the neighbor overhearing, we also propose routing set middleware, for which we

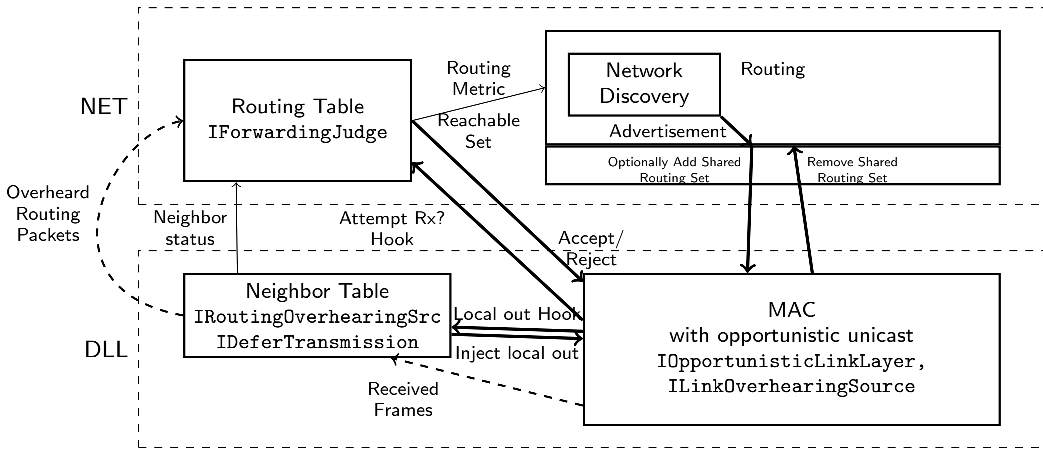

Figure 2: OR cross layer interfaces for network discovery, packet deferral, overhearing and acceptance.

use the `TlvOption` specification as a starting point. This piggybacks the routing set into a proportion of packets, and hence reduces the need for extra advertisement packets. The routing set overhearing is handled by listening to `IRoutingOverhearingSource`.

3.1 Implementation of existing protocols

The structure of the above proposals comes from our experience implementing the opportunistic RPL (ORPL) protocol [5], and studying techniques to improve neighbor discovery and packet throughput. For example, our naive neighbor discovery protocol could be replaced by Find [3], a discovery protocol specifically tuned for intermittent devices. Likewise, neighbor prediction could benefit from an advanced predictor tailored for intermittent networks [4], which would implement `IDeferTransmission`.

The opportunistic operation of ORPL can be seen in Figure 1 demonstrating the variety of routes taken, where each next-hop decision depends on the instantaneous availability of forwarding nodes. This implementation makes use of the proposed interfaces to encourage comparisons to new protocols.

3.2 Working Implementation using Opportunistic Interfaces

The implementation is based on opportunistic routing for WSN (ORW)[6] and implements the described interfaces. ORW uses neighbor encounter detection to calculate and expected cost (in terms of node duty cycle) to the sink, termed EDC. Our implementation implements both the neighbor table, MAC forwarder negotiation protocol and the network layer routing protocol. While the implementation does not yet exactly match our proposed interfaces, we are working towards this. The source code is available at https://github.com/UoS-EEC/INET-opportunistic-routing, and is briefly discussed here.

The MAC protocol `ORWMac` implements the `IOpportunisticLinkLayer` interface, where the accept reject interface is inherited from `inet::NetFilterBase`. However, the preferred interface could use a query out and accept in gates passing the received packet temporarily to the routing layer. The signals emitted when the link layer overhears an incoming transmission are categorized depending on if it was expected, for example an acknowledgement response to a transmission, or coincidental, for example an initial packet reception or advertisement packets. Also, a signal is emitted at the end of a period where packets are expected, regardless of whether they are actually received.

The routing table `ORWRoutingTable` provides the `calculateUpwardsCost(L3Address dest)` interface, for the `ORWRouting` protocol to tag outgoing packets. For incoming packets it implements the

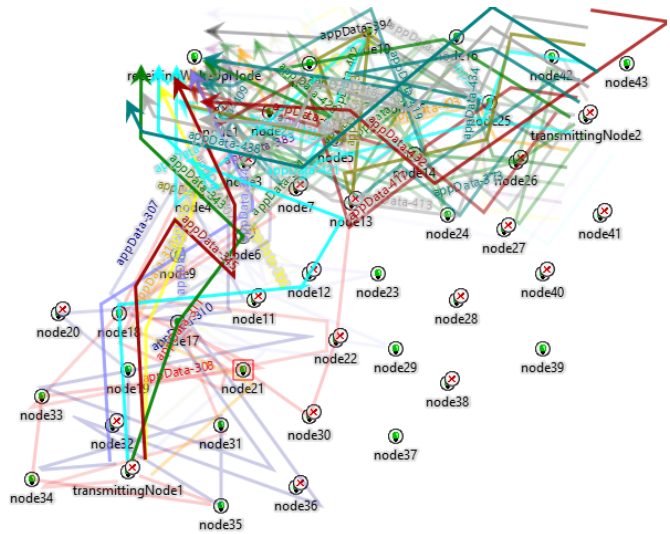

Figure 3: Operation of Many hop ORW demonstration with intermittently-powered forwarding nodes, showing variety of routes taken from two transmitting nodes to single destination.

other half of the forwarding request interface with `inet::NetfilterBase::HookBase` and likewise the preferred interface could use message gates implementing `IForwardingJudge`. The advantage of using the hooks is that the containing `NetworkLayerNodeBase` and `IWirelessInterface` do not need to be modified. A demonstration of upward routing ORW can be seen in Figure 3.

We extend the `ORWRouting` class to enable downward routing implementing ORPL [5] in `ORPLRouting` and `ORPLRoutingTable`. This allows any node in the network to be reachable from any other node and requires occasional routing set sharing. The routing set is added to outgoing packets after routing has happened and removed before routing layer acceptance by `ORPLRouting`, the `ORPLRoutingTable` listens to `ILinkOverhearingSource` for updating routing set information.

Since there is not currently a distinction between the neighbor and routing table and because deferred transmissions are not implemented, `IRoutingOverhearingSource` and `IDeferTransmission`, are currently unused, but are useful elements that can be used to improve the OR implementation.

4 Conclusion

OR can route information in intermittently-powered WSNs and to characterize their performance, in consideration of the power consumption, we have developed interfaces to model them. There are several techniques that purport to improve certain aspects such as neighbor availability prediction, or the routing itself. The proposed interfaces between modules allows for innovation and optimization of these narrow aspects of intermittent networking to be tested in a whole system, as well as against each other. Our implementation enables testing of opportunistic protocols intermittently-powered networks to explore suitable solutions and parametric optimisation.

References

[1] C. Renner and M. Zella. "The Internet of Intermittent Things, a Land of Low-Hanging Fruits". In: *Proc. of the 7th Int. Workshop on Energy Harvesting & Energy-Neutral Sensing Systems*. ENSsys'19. New York, NY, USA: ACM, Nov. 2019, pp. 49–51. ISBN: 978-1-4503-7010-3. DOI: 10.1145/3362053.3363493.

[2] A. Udugama et al. “Opportunistic Networking Protocol Simulator for OMNeT++”. en. In: *Proc. of the 4th OMNeT++ Community Summit*. arXiv: 1709.02210. Sept. 2017.

[3] K. Geissdoerfer and M. Zimmerling. “Bootstrapping Battery-free Wireless Networks: Efficient Neighbor Discovery and Synchronization in the Face of Intermittency”. en. In: *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*. 2021, pp. 439–455.

[4] M. M. I. Rajib and A. Nasipuri. “Predictive Retransmissions for Intermittently Connected Sensor Networks with Transmission Diversity”. In: *ACM Trans. Embed. Comput. Syst.* 17.1 (Sept. 2017), 12:1–12:25. ISSN: 1539-9087. DOI: 10.1145/3092947.

[5] S. Duquennoy, O. Landsiedel, and T. Voigt. “Let the tree Bloom: scalable opportunistic routing with ORPL”. en. In: *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems - SenSys ’13*. Roma, Italy: ACM Press, 2013, pp. 1–14. ISBN: 978-1-4503-2027-6. DOI: 10.1145/2517351.2517369.

[6] E. Ghadimi et al. “Opportunistic Routing in Low Duty-Cycle Wireless Sensor Networks”. In: *ACM Trans. Sen. Netw.* 10.4 (June 20, 2014), 67:1–67:39. ISSN: 1550-4859. DOI: 10/f22n3b.