Author manuscript, published in "19th International Conference on Software, Telecommunications and Computer Networks, Split : Croatia (2011)"

hal-00651603, version 1 - 14 Dec 2011

The P2P-RPL Routing Protocol for IPv6 Sensor Networks: Testbed Experiments

Emmanuel Baccelli, Matthias Philipp
INRIA Saclay, France
E-mail: name.lastname@inria.fr
Mukul Goyal
UWM, USA
E-mail: mukul@uwm.edu

Abstract

An important part of the foreseen Internet of Things consists in wireless sensor networks running adapted IPv6 protocols. Since the way sensors are scattered is generally unplanned and may evolve with time, a routing protocol is needed in order to provide paths across such networks. Efforts towards standardizing RPL, a routing protocol targeting sensor networks, have thus recently taken place. This paper analyzes some fundamental tradeoffs inherent to RPL, which enables the protocol to require smaller routing state than most other routing protocols. However, these tradeoffs are on the other hand an issue in several Home and Building Automation use-cases, which require sensor to sensor communication – aside of communication from sensor to sink. RPL basically requires that all communication paths go through a central router (the sink), which provides severely suboptimal paths in these use-cases. In order to alleviate this, an extension of the protocol is proposed based on a reactive scheme that can provide shorter paths on-demand, without necessarily going through the sink. This paper then evaluates this extension via experiments on a sensor network testbed running RPL and its extension over IEEE 802.15.4 radio. These experiments confirm that the extension provides substantially shorter paths.

1. Introduction

Sensor networking is a key element of the future Internet of Things, a substantial part of the billions of communicating machines that are soon to blend in the global IP network, including devices ranging from actuators to home appliances, from smart meters to smart dust. Sensors are devices used for distributed and automated monitoring of various parameters such as temperature, movement, noise or radioactivity levels etc., depending on the type of sensor.

While some of these sensors will connect to the network via wire or power line communication, many sensors will use radio communications. Typically, a number of such devices, identical to one another, are scattered in the zone to be monitored. Each sensor then monitors the parameters to be measured in its vicinity and communicates through its single radio interface with its peers, spontaneously creating a wireless network. Using this network, sensors self-organize distributed computations or information gathering at a central control point, which is generally called the *sink* in this context.

From a networking point of view, scattered sensors may be more or less remote from one another – a given sensor may for instance require some peers to forward information towards the sink, because the sink is outside of its radio range. Appropriate network protocols are thus required to enable each sensor to, on one hand, *directly communicate* with peers that are within its radio range, and on the other hand, *indirectly communicate* with other devices that are only reachable through some peer with which the sensor can directly communicate, as shown in Fig. 1.

Several different technologies can be used to enable direct wireless sensor communication at the link layer (layer 2), including IEEE standards such as 802.15.4, Bluetooth, or low-power Wi-Fi, and future technologies yet to come.

The way sensors are scattered is generally unplanned (aside of the central role of the sink). Therefore, other mechanisms are used to enable indirect wireless sensor communication at the network layer (layer 3), i.e. routing protocols for multi-hop wireless sensor networks, which are the focus of this paper. The IETF [14] is currently elaborating such a routing protocol, RPL (Routing Protocol for Low power and lossy networks [1]), which is supposed to become the standard routing protocol for IPv6-based multi-hop wireless sensor networks. While RPL focuses on establishing routes between a special sink router and all remaining routers, the RPL point-to-point (P2P) extension is targeted towards creating routes between arbitrary routers.

1.1. Related Work

Several recent papers have studied the performance of RPL in various scenarios based on simulations. [3] presents

hal-00651603, version 1 - 14 Dec 2011

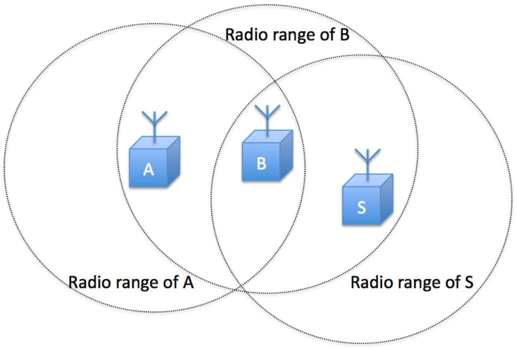

Figure 1. Indirect wireless communication.

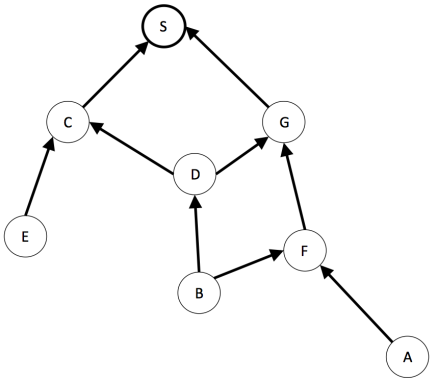

Figure 2. Destination Oriented Directed Acyclic Graph (DODAG) rooted at router S.

results on path stretching with RPL and [4] studied RPL convergence time and network churn. Other efforts, such as [5] or [6] studied network flooding optimization based on the routing structure provided by RPL. RPL was also evaluated in smartgrid environments via simulations in [7] or in [8]. Available open-source code bases have been developed within the main sensor OS platforms, such as Contiki [9] and TinyOS [11]. This paper presents on the other hand results obtained with experiments using RPL in an actual testbed, and a source-initiated reactive extension of the RPL protocol called P2P-RPL that drastically improves the performance of the protocol in Home and Building automation scenarios, by providing much shorter paths between sensors. Section 2 will overview RPL characteristics and tradeoffs, before Section 3 introduces the reactive extension of the protocol, targeting Home and Building Automation use-cases. Section 4 will then present the testbed and the experimental results obtained with these protocols, before we conclude on the matter.

2. RPL: Routing Protocol for Low Power Lossy Networks

RPL [1] is a routing protocol that organizes routers along a Destination Oriented Directed Acyclic Graph (DODAG), a category of Directed Acyclic Graph [12], rooted at the sink (router S in Fig. 2). The DODAG root initiates the DODAG formation by periodically originating DODAG Information Object (DIO) messages which it advertises via link-local multicast. DIO messages carry information such as the DODAG root's identity, the routing metrics in use, as well as the originating router's depth (called the ”rank”) in the DODAG. A router joins the DODAG taking in consideration these factors, determines its own rank in the DODAG based on the information advertised by its neighbors in their DIOs. The router chooses as parent(s) in the DODAG the neighbor(s) through which its resulting rank is the smallest, amongst neighbors advertising DIO messages. Once a router has thus joined the DODAG, it has a path to the sink through its parent(s), and the router can then originate its own DIO messages.

RPL thus provides paths from routers to a sink while requiring routers to store very little forwarding and routing table information, essentially information about its parent in the DODAG. This characteristic is compatible with wireless sensors, which are typically cheap and rather unreliable devices that have drastic constraints in terms of CPU and memory (typically a few kilobytes of RAM and ROM in total).

The transmission rate of DIO messages furthermore follow a Trickle [10] policy which aims at pruning unnecessary transmissions by monitoring information consistency between neighbors. When a node's data does not agree with its neighbors, that node communicates quickly to resolve the inconsistency. When nodes agree, they slow their communication rate exponentially, such that nodes send packets very infrequently. This characteristic is also compatible wireless sensors' portable power supply capacities, which are drained too fast if they need to transmit too often.

2.1. RPL Tradeoffs

In order to keep the size of forwarding and routing tables small, RPL does not by default provide paths from the sink back to sensors. However, the availability of paths from the sink to individual sensors are necessary in many scenarios, including industrial actuators and selective sensor queries [19]. In order to address this lack, an RPL router that requires a path from the sink to itself must send a Destination Advertisement Object (DAO) message upwards along the DODAG all the way up to the root, which records and install this path. The DAO mechanism can be operated either in storing or in non-storing mode. In storing mode each router needs to store routing information in order to forward packets hop-by-hop. In contrast, the non-storing mode employs source-routes which are managed at the sink

hal-00651603, version 1 - 14 Dec 2011

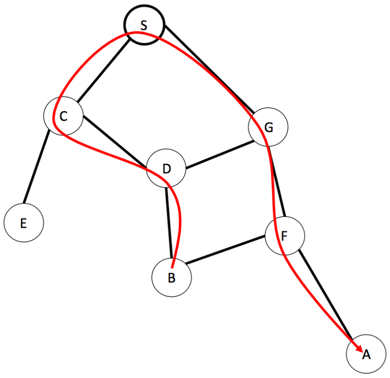

Figure 3. Communication between router B and router A through the sink, provided by RPL.

in order to further reduce the size of forwarding tables on the other routers.

Similarly, in order to keep the size of forwarding and routing tables small, RPL does not by default provide paths between arbitrary sensor pairs. Such paths are however needed in several scenarios, including home and automation use cases [18] [17]. A simple example a remote control (or a motion sensor) that suddenly needs to communicate with a lamp module, whose network address it knows a priori. In order to address this lack, an RPL router A that requires a path from another router B to itself must send a Destination Advertisement Object (DAO) message upwards along the DODAG to establish a path from the sink to router A, and router B can then communicate with router A via the first ancestor common to router A and B in the DODAG that has a path to A – at worse, via the sink (see Fig. 3). If RPL is operated in non-storing mode, the worst case happens systematically: all communications are via the sink, which is the only router in the network to store source routes to other nodes.

As shown in [3], RPL provides paths that are thus often much longer than the shortest available paths. Moreover, the constraint to route only along a DODAG may potentially cause severe traffic congestion near the DODAG root. Finally, the constraint for every possible destination in the DODAG to originate a DAO is problematic because it is a proactive destination-initiated process which is not compatible with many Home and Building Automation scenarios, such as the simple example of a remote control suddenly needing to communicate with a lamp module – a fundamentally a reactive source-initiated process.

For these reasons, P2P-RPL [2] provides a reactive mechanism that establishes source-initiated discovery of sensor to sensor paths that are not necessarily along the DODAG. This mechanism is introduced in the following section.

3. P2P-RPL: Reactive Discovery of Point-to-Point Routes with RPL

RPL provides dog-legged paths for point to point (P2P) communication between arbitrary sensors in the network, as described above in Section 2. Since P2P communication is a fundamental requirement for several applications, including some in Home and Building Automation, an extension of the protocol, called P2P-RPL [2] is currently being considered in order to provide shorter P2P paths between sensors, when available.

3.1. Concept

P2P-RPL allows routers to discover and establish path(s) to another router, based on a simple reactive mechanism. Essentially, when a router S needs to discover a path to another router D, router S originates a message similar in functionality to an AODV Route-Request [16] indicating it seeks a path to S. This message is piggy-backed on DIO messages, and disseminated throughout the network using Trickle, effectively creating a temporary DODAG rooted in S. While traveling across the network, the message installs temporary next-hop information towards S on the traversed routers, and may accumulate information about the path travelled so far. Upon receiving such a message, router D sends a message back to S, similar in functionality to an AODV Route-Reply, along the recorded path, thus establishing a path between S and D, and the temporary DODAG eventually expires.

P2P-RPL uses the same mechanisms as basic RPL to form the DODAG. It introduces a new DIO option that specifies the address that should be discovered and records the traversed path. The lifetime of the DODAG is restricted to the time of the route request. P2P-RPL allows to use source routes as well as hop-by-hop routes and it is possible to specify metric constraints for the discovered routes.

3.2. Implementation

In order to study the behavior of RPL and P2P-RPL in vivo, the specification has been implemented on Contiki [13], an operating system for wireless sensor networks used and actively developed by a wide industrial and academic community. Contiki was initially chosen because it includes an IPv6 stack with 6LoWPAN support, as well as ContikiRPL, an implementation of basic RPL, which was used as basis for our P2P-RPL implementation.

4. Experiments with P2P-RPL

While previous work on P2P-RPL was based on simulation and focussed on theoretical aspects [3], [4], this paper analyzes the protocol based on experiments on a real-world WSN deployment.

hal-00651603, version 1 - 14 Dec 2011

4.1. Testbed

Experiments have been carried out on the Senslab [15] testbed. The testbed for wireless sensor networks comprises four sites in France which each offer 256 MSP430-based sensor nodes. The nodes are equipped with different radio technologies. While some of them are mobile, the majority of the nodes is fixed. At the site in Lille, which has been used for the experiments, the nodes are deployed on an area of 5x5 meters and positioned in a three-dimensional grid topology with a distance of 60 cm between nodes. A photo of the testbed is presented in figure 4.

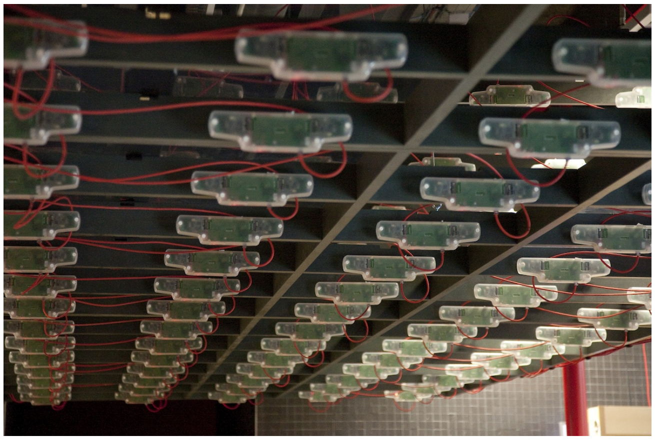

Figure 4. Sensors at the Senslab site in Lille

For the experiments presented in this paper, a subset of 27 fixed nodes operating at 2.4 GHz with IEEE 802.15.4 radio interfaces have been used. The resulting network topology with an average node degree of 4.39 and a resulting DODAG are shown in figures 5 and 6 respectively. Sensors were running the Contiki operating system including IPv6 stack, 6LowPAN, and P2P-RPL as described in Section 3.2.

4.2. Experiments

The first set of experiments focused on path length with P2P-RPL on one hand and with plain RPL on the other hand. Random pairs of nodes were successively chosen as source and target of an arbitrary data transfer. With P2P-RPL, a route discovery to the target was initiated by the source node and the returned route was recorded. For plain RPL, a global DAG was used to establish upward routes and the DAO mechanism in non-storing mode was used to provide downward routes in order to provide paths from source to destination as described in Section 3, and the path through the sink was thus recorded. Such experiments were repeated 230 times on the same set of nodes for each protocol. The DODAG that was formed during the experiments is shown in figure 6. Note that this is only a logical representation: nodes depicted next to each other are not necessarily physical neighbors, unless they are displayed as linked with an edge in the figure.

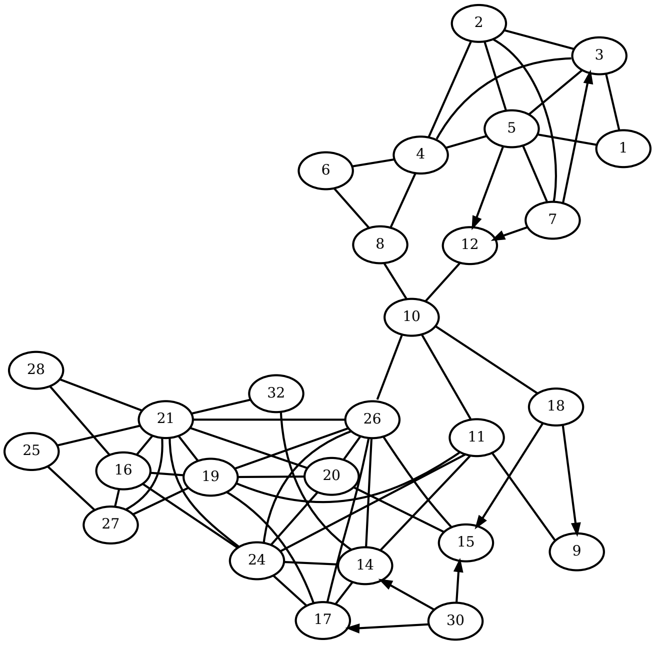

Figure 5. Network topology during the experiments.

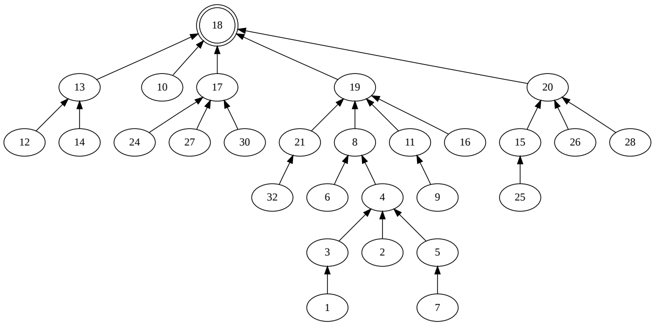

Figure 6. DODAG that formed during the experiments.

The resulting route lengths are presented in figure 7. While data packets in plain RPL need to traverse almost 5 links in average, the average route length is approximately half when using P2P RPL on the same network. Note that this factor of 2 is a rather good case, as with other topologies yielding a deeper DODAG, plain RPL provides much longer paths: a factor of 15 is for instance predicted with the topology studied in [3].

The second objective of P2P-RPL is to decrease traffic density near the DODAG root. Figure 8 shows how many routes traverse the root node with P2P-RPL compared to that with plain RPL. Since in RPL non-storing mode all routes traverse the root node, the figure depicts the results for storing mode, where routes traverse the common ancestor, which is thus the mode that is most favorable to RPL in

hal-00651603, version 1 - 14 Dec 2011

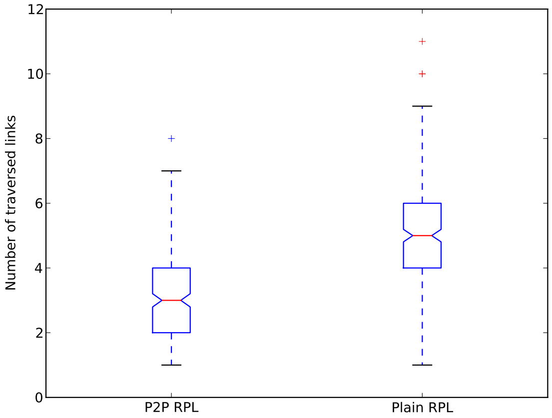

Figure 7. Route length for P2P-RPL and plain RPL.

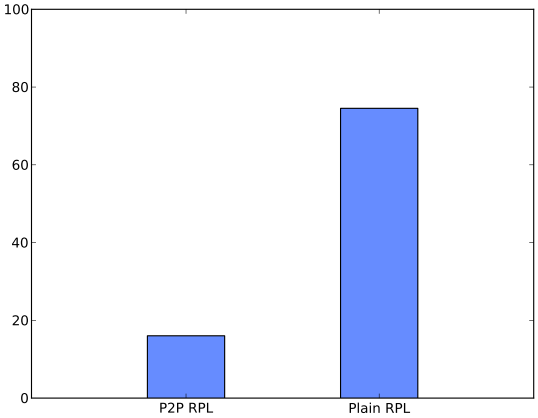

Figure 8. Percentage of routes traversing the DODAG root.

that regard. It can be nevertheless be seen that even in this case 74.53% of all routes traverse the DODAG root, while in P2P RPL this is only the case for 16.03% of the routes.

We also noted that the time between originating the route request and receiving the route reply was measured to be 8.43 seconds in average. However, to allow for radio duty cycling, Contiki uses a value of 4096 ms for the minimum DIO interval instead of 8 ms as recommended by the specification. Further experiments should evaluate the influence of trickle timer variables on the duration of the route request.

The experiments further showed that although 99.16% of the route requests reached their target, only 57.87% of the route reply messages arrived at the origin. This is attributed to the absence of any redundancy or reliability mechanism for route reply messages: a single reply packet gets easily lost in multi-hop wireless networks. This issue should be addressed in further versions of the specification.

5. Conclusion

This paper briefly analyzed RPL, the routing protocol targeting sensor networks currently standardized by the IETF within the IPv6 protocol stack. In order to accommodate the typical memory constraints of these devices, the tradeoffs made with RPL enable the protocol to function with very small routing state. However, on the other hand, these tradeoffs provide significantly suboptimal paths in several Home and Building Automation use-cases. The paths provided by RPL from an arbitrary source to an arbitrary destination in the sensor network must basically go through the sink, which provides dog-legged routes even if short-cuts exist. An extension of RPL, called P2P-RPL, is thus proposed to discover such short-cuts when available, based on a simple reactive, on-demand approach.

This paper studied the performance of RPL and P2P-RPL on a testbed of sensor devices using IEEE 802.15.4 radios. The experiments confirmed that P2P-RPL establishes considerably shorter paths than plain RPL and reduces traffic density at the DODAG root. Shorter paths are expected to increase data packet delivery ratio and decrease delays because less hops are necessary along the path, while overall network life-time should increase because less devices need to retransmit. Such hypothesis will be verified via further experiments, as differences in RPL and P2P-RPL control traffic patterns may also have an influence on energy drain, and different data traffic patterns will also yield different results in terms of routing protocol efficiency. The results of such experiments with RPL and P2P-RPL, in a real-world environment, also help identify shortcomings of the current specifications, which should be addressed in future versions of the specifications currently being standardized.

References

[1] T. Winter, P. Thubert et al. “RPL: IPv6 Routing Protocol for Low power and Lossy Networks,” IETF Internet Draft draft-ietf-roll-rpl-19 (work in progress), March 2011.

[2] M. Goyal, E. Baccelli, A. Brandt, R. Cragie, J. Martocci, “Reactive Discovery of Point-to-Point Routes in Low Power and Lossy Networks,” IETF Internet Draft draft-ietf-roll-p2p-rpl-03 (work in progress), May 2011.

[3] W. Xie, M. Goyal, H. Hosseini, J. Martocci, Y. Bashir, E. Baccelli, A. Durresi, “A Performance Analysis of Point-to-Point Routing along a Directed Acyclic Graph in Low Power and Lossy Networks,” Proceedings of the 13th International Conference on Network-Based Information Systems (NBiS), Takayama, Japan, September 2010.

hal-00651603, version 1 - 14 Dec 2011

[4] W. Xie, M. Goyal, H. Hosseini, J. Martocci, Y. Bashir, E. Baccelli, A. Durresi, “Routing Loops in DAG-based Low Power and Lossy Networks,” Proceedings of the IEEE Advanced Information Networking and Applications (AINA), Perth, Australia, April 2010.

[5] T. Clausen, U. Herberg, “Comparative Study of RPL-Enabled Optimized Broadcast in Wireless Sensor Networks,” Proceedings of the 6th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), December 2010.

[6] T. Clausen, U. Herberg, “Multipoint-to-Point and Broadcast in RPL,” Proceedings of the 13th International Conference on Network-Based Information Systems (NBiS), Takayama, Japan, September 2010.

[7] J. Tripathi, J.C. de Oliveira, J.P. Vasseur, “Applicability Study of RPL with Local Repair in Smart Grid Substation Networks,” Proceedings of the 1st IEEE International Conference on Smart Grid Communications (SmartGridComm), Gaithersburg, USA, October 2010.

[8] D. Wang, Z. Tao, J. Zhang, A. Abouzeid, “RPL Based Routing for Advanced Metering Infrastructure in Smart Grid,” Proceedings of the IEEE International Communications Conference (ICC), Cape Town, South Africa, May 2010.

[9] N. Tsiftes, J. Eriksson, N. Finne, A. Dunkels et al., “A framework for low-power IPv6 routing simulation, experimentation, and evaluation,” Proceedings of ACM SIGCOMM, New York, USA, October 2010.

[10] P. Levis, T. Clausen, J. Hui, O. Gnawali, J. Ko, “RFC 6206: The Trickle Algorithm,” Internet Engineering Task Force (IETF) Request For Comments, http://ietf.org/rfc/rfc6206.txt, March 2011.

[11] J. Ko, O. Gnawali, D. Culler, S. Dawson-Haggerty, A. Terzis, “Evaluating the Performance of RPL and 6LoWPAN in TinyOS,” Proceedings Workshop on Extending the Internet to Low power and Lossy Networks (IP+SN), Chicago, USA, 2011.

[12] J. Martin, “Distribution of the time through a directed acyclic network,” Oper. Res., vol. 13, pp. 44-66, 1965.

[13] The Contiki Operating System, http://www.sics.se/contiki/

[14] The Internet Engineering Task Force, http://www.ietf.org

[15] Senslab: Very Large Scale Open Wireless Sensor Network Tesbed, http://www.senslab.info/

[16] C. E. Perkins, E. M. Royer, S. R. Das, “RFC 3561: Ad Hoc On-Demand Distance Vector Routing,” Internet Engineering Task Force (IETF) Request For Comments (experimental), http://ietf.org/rfc/rfc3561.txt, July 2003.

[17] A. Brandt, J. Buron, G. Porcu, “RFC 5826: Home Automation Routing Requirements in Low-Power and Lossy Networks,” Internet Engineering Task Force (IETF) Request For Comments, http://ietf.org/rfc/rfc5826.txt, April 2010.

[18] J. Martocci et al., “RFC 5867: Building Automation Requirements in Low-Power and Lossy Networks,” Internet Engineering Task Force (IETF) Request For Comments, http://ietf.org/rfc/rfc5867.txt, June 2010.

[19] K. Pister, P. Thubert et al., “RFC 5673: Industrial Routing Requirements in Low-Power and Lossy Networks,” Internet Engineering Task Force (IETF) Request For Comments, http://ietf.org/rfc/rfc5673.txt, October 2009.