Multipoint-to-Point and Broadcast in RPL

Thomas Clausen, Ulrich Herberg
Laboratoire d'Informatique (LIX) – Ecole Polytechnique, France
Email:Thomas@ThomasClausen.org, Ulrich@Herberg.name

***Abstract*—Recent trends in Wireless Sensor Networks (WSNs) have suggested converging to such being IPv6-based. To this effect, the Internet Engineering Task Force has chartered a Working Group to develop a routing protocol specification, enabling IPv6-based multi-hop Wireless Sensor Networks. This routing protocol, denoted RPL, has been under development for approximately a year, and this paper takes a critical look at the state of advancement hereof: it provides a brief algorithmic description of the protocol, and discusses areas where – in the authors view – further efforts are required in order for the protocol to become a viable candidate for general use in WSNs. Among these areas is the lack of a proper broadcast mechanism. This paper suggests two such broadcast mechanisms, both aiming at (i) exploiting the existing routing state of RPL, while (ii) requiring no additional state maintenance, and studies the performance of RPL and of these suggested mechanisms.**

I. Introduction

The general context for routing in Wireless Sensor Networks (WSNs) is small, cheap devices whose primary function is data acquisition, and for which communications capabilities are a "commodity to their primary function" – a necessary, but in preference unobtrusive, functionality, specifically targeted to the precise goal which the WSN is deployed to satisfy. As an example, a WSN deployed for environmental monitoring might contain a set of temperature sensors, sending "notifications" to a central controller when the temperature exceeds certain thresholds – and occasional "keepalive" messages otherwise, to let the controller know that the sensors are still operational. Traffic from the controller to the individual sensors may be limited to "setting the thresholds" – possibly rarely, such as at system deployment, or even never such as would be the case with factory set thresholds.

A. WSN Traffic Flows

The communications requirements for WSNs are in contrast to "traditional networks", wherein communications devices (network interfaces, switches, routers) have carrying data traffic as their sole raison d'être, and in which devices do not make any a-priori assumptions such as the characteristics of the traffic they will be carrying. WSNs assume an a-priori knowledge of the traffic patterns to optimize for – with sensor-to-controller traffic (*multipoint-to-point*) being predominant, controller-to-sensor traffic (*point-to-multipoint*) being rare and sensor-to-sensor traffic being somewhat esoteric[1].

[1]Note that while this may be commonly assumed, this is not a universal distribution of traffic patterns in WSNs – there are scenarios in which sensor-node to sensor-node traffic is assumed a more common occurrence, such as [1].

B. WSN Trade-off's

Low-power consumption, minute physical sizes, low price-points and ruggedness against the environment are among the industrial or commercial keywords, often associated with wireless sensors – and which entail challenging constraints (in terms of the computational power, permanent and temporary storage and in the characteristics (capacity) of the wireless interfaces) for designing routing algorithms. WSN routing protocols are therefore inherently compromises: trade-offs are made in adapting to the specific constraints under which they are to operate – the first of these is usually "generality". WSN routing protocols generally and narrowly consider only the traffic characteristics of their target environment as "valid", and discard all other traffic characteristics in the name of satisfying operational constraints; two of the most common such constraints brought forward are strict bounds on in-router state and on control traffic. A second trade-off is often in route optimality: stretched (non-optimal) routing paths are an acceptable trade-off for lower control traffic from a routing protocol, with the hypothesis that traffic flows will be such that the impact of such stretched paths will be negligible.

The perceived optimal routing protocol might thus be described as a routing protocol which requires zero in-router state and zero control traffic overhead, while providing non-stretched routing paths. Such a protocol is possible, although may not be desirable.

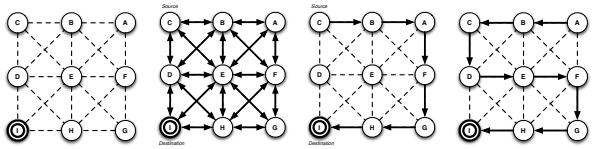

Figure 1. "Route Stretching" vs "number of transmissions"

Consider the example in figure 1a. The network connectivity is as indicated by the dotted lines, with the source and destination indicated by *Source* (node C) and *Destination* (node I), respectively. A "perceived optimal routing protocol" would be, as illustrated in figure 1b, simply flooding data traffic. Such entails no control traffic overhead and no in-router state, and a packet from C will arrive at I via a path of length 2 (*i.e.* a routing path stretch of 1). Data transmission between C and I via a path such as the one indicated in figure 1c appears intuitively better. While the routing path stretch is 3 (6 hops), at least routers D and E do not retransmit. An even worse situation is possible, as illustrated in figure 1e: all routers still

retransmitting and receiving as many copies of a packet as in flooding – but with a routing path stretch of 4.

A flooding operation, as in figure 1b, would in this case entail 8 transmissions (*i.e.* $(n-1)$ transmissions, with n being the number of nodes in the network) – just as "bad"[2] for battery consumption and media occupation as if the path length had been of 8, as in figure 1e. On the other hand, the routing path in figure 1e did not appear "by magic": a (more or less optimal) routing protocol has provided this path, and in order to do so generated a certain amount of control traffic.

As a measure of success, "routing path stretch" is an inappropriate metric, when used alone. In deployments with heavy unicast traffic, it might be reasonable to trade off more state and more control traffic in order to obtain shorter paths, whereas in scenarios where such unicast traffic is light, a longer path may be a reasonable trade-off in order to reduce state and control traffic. If in a network unicast traffic is both light and rare, simple flooding, and so trading off "route stretching" (or, more appropriately, "total number of transmissions on the wireless medium in order to successfully deliver the data packet at the destination") and state for simpler logic in the router and no control traffic, might be reasonable, as might flooding be reasonable if the majority of traffic is (very light) broadcast.

C. Paper Outline

The remainder of this paper is organized as follows: section II provides an overview of the activities of the IETF ROLL working group, chartered to develop routing protocols for IPv6-based sensor networks, as well as provides a description and critical discussion of the RPL routing protocol, developed within that working group. RPL provides relatively well defined and well understood support for multipoint-to-point traffic – and is currently developing mechanisms for supporting point-to-multipoint traffic as well. Section III suggests a couple of different mechanisms for providing also support for broadcast traffic in a WSN, by way of using the data structures and topologies already maintained by RPL. Section IV provides a performance study of the multipoint-to-point performance of RPL, as well as a comparative study of the suggested broadcast mechanisms. Section V concludes this paper.

II. State of the art: ROLL and RPL

ROLL is the abbreviation of an IETF Working Group named "Routing Over Low power and Lossy networks". This working group has as objective to develop a routing protocol for WSN-like networks, based on IP.

The unofficial goal, which this Working Group tries to attain, is to prevent fragmentation in the WSN market by providing an IP-based routing standard and solicit broad industrial support behind that standard. To this end, the Working Group is operating with a very tight schedule and an objective of completing the standardization effort early 2010, satisfying only whatever requirements have been expressed within that time-frame.

The current proposal by the ROLL Working Group is denoted "Routing Protocol for Low Power and Lossy Networks" (RPL), an early draft version hereof exists [2]. The objective of this protocol is to target networks which *"comprise up to thousands of nodes"*, where the majority of the nodes have very constrained resources, where the network to a large degree is "managed" by a (single or few) central "supernodes", and where handling mobility is not an explicit design criteria. Supported traffic patterns include multipoint-to-point, point-to-multipoint and point-to-point traffic. The emphasis among these traffic patterns is to *optimize for* multipoint-to-point traffic, to *reasonably support* point-to-multipoint traffic and to *provide basic features for* point-to-point traffic, in that order.

The basic construct in RPL is the DODAG — a destination oriented DAG, rooted in a "controller", in figure 2. In the converged state, each WSN router has identified a stable set of parents, on a path towards the "root" of the DODAG, as well as a *preferred parent*. Each router, which is part of a DODAG (*i.e.* has selected parents) will emit *DODAG Information Object* (DIO) messages, using link-local multicasting, indicating its respective *Rank* in the DODAG (*i.e.* their position – distance according to some metric(s), in the simplest form hop-count – with respect to the root). Upon having received a (number of such) DIO messages, a router will calculate its own rank such that it is greater than the rank of each of its parents, and will itself start emitting DIO messages. Thus, the DODAG formation starts at the root, and spreads gradually to cover the whole network.

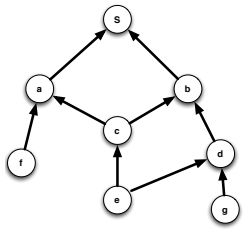

Figure 2. RPL Basic Construct: DODAGs

As a Distance Vector protocol, RPL [2] contains rules restricting the ability for a router to change its rank. Specifically, a router is allowed to assume a smaller rank than previously advertised (*i.e.* to logically move closer to the root) if it discovers a parent advertising a lower rank (and it must then disregard all previous parents with higher ranks), while the ability for a router to assume a greater rank (*i.e.* to logically move farther from the root) in case all its former parents disappear, is restricted to avoid count-to-infinity problems. The root can trigger "global recalculation" of the DODAG by way of increasing a sequence number in the DIO messages.

A. RPL Data Traffic Flows

The DODAG so constructed is used for installing routes in the WSN routers: the "preferred parent" can serve as a

[2]Actually, even worse: in order to prevent "looping" packets, state would have to be maintained in each sensor node, ensuring that each such packet would be retransmitted no more than once.

default route towards the root, or the root can embed in its DIO messages the destination prefixes, also included by DIOs generated by WSN routers through the WSN, to which it can provide connectivity. Thus, RPL provides "upward routes" or "multipoint-to-point routes" from the sensors towards the controller.

"Downward routes" are installed by having the sensors issue *Destination Advertisement Object* (DAO) messages, which propagate via parents towards the routes, and which describe which prefixes belong to, and can be reached via, which WSN router. Each intermediate WSN router, forwarding a DAO message towards the root, adds its address to a *reverse routing stack* in the DAO message, thereby providing the source with the ability to do source routing for reaching addresses in the WSN.

Sensor-to-sensor routes are as default supported by having the source sensor transmit via its default route to the root, which will add a source-route to the received data for reaching the destination sensor.

B. RPL Operational Requirements

The minimal set of in-router state, required in a WSN router running RPL is, (i) the identifier of the DODAG root, (ii) the address and rank of the preferred parent, (iii) the configuration parameters shared by the DODAG root (notably, destination prefixes and message emission timers) and (iv) the maximum rank that the WSN router has itself advertised. For redundancy, a WSN router running RPL can maintain information describing additional parents (up to and including all its parents), which may allow rapidly changing its preferred parent (and thus its "next hop") in case the former preferred parent becomes unreachable.

RPL message generation is timer-based, with the root able to configure suitable back-off of message emission intervals using *trickle timers* [3].

C. RPL Discussion

In its basic form, RPL is a fairly simple-to-understand and simple-to-implement distance-vector protocol. The DODAG formation mechanism, using DIO messages, is currently well understood, and despite the specification hereof in [2] remaining somewhat ambiguous, the authors of this paper managed to develop and test an implementation "from scratch" within about a week.

The DODAG formation mechanism is not without potential issues, however. First, parents (and the preferred parent) are selected based on receipt of DIO messages, without verification of the ability for a WSN router to successfully communicate with the parent – *i.e.* without any bidirectionality check of links. In a wireless environment, unidirectional links are no rare occurrence, and can simply happen as illustrated in figure 3: the gray device, *X*, illustrates a source of environmental interference, preventing route *b* from successfully receive transmissions from *a*. This may, however, not prevent *b* from transmitting DIOs, received by *a* and which may contain

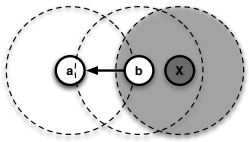

Figure 3. Unidirectional link due to radio interferrence

information causing *a* to select *b* as both parent and preferred parent.

As *b* is a "useless" next-hop for *a*, due to the interference from *X*, this is a bad choice. RPL suggests using Neighbor Unreachability Detection (NUD) [4] to detect and recover from this situation, when it occurs that *a* tries (and fails) to actually use *b* for forwarding traffic. NUD is based upon observing if a data packet is making forward progress towards the destination, either by way of indicators from upper-layer protocols (such as TCP), from lower-layer protocols (such as Link Layer ACKs) or – failing these two – by unicast probing. A couple of problems can be noted regarding this approach.

First, absent all WSN routers consistently advertising their reachability through DAO messages, a protocol requiring bi-directional flows between the communicating devices, such as TCP, will be unable to operate. Even if such bi-directional flows are enabled, the source detecting, by way of an upper layer protocol, that no forward progress is possible, is of restricted use: the source can not know if it is its "preferred parent" (next hop) which is unreachable, or if it is a problem further along the path (even outside the WSN). Thus, any corrective action that the source might take (changing preferred parent, moving to a higher rank within the limits allowed, etc.) may be unable to alleviate the problem, and corrective actions may even be counter-productive (poison the sub-dag, for example).

Second, there is a change that the radio range of a unicast (as would be used for data delivery via the next hop towards the root) would differ from the radio range of DIOs, which are sent using link-local multicast[3].

Third, upon having been notified by NUD that the "next hop" is unreachable, a WSN router must discard the preferred parent and select another preferred parent – hoping that this time, the preferred parent is actually reachable.

Fourth, the selection of parents and preferred parent is based on receipt of DIO messages only, and is based on the rank of the candidate parents. Absent other complementary mechanisms (which are currently not specified as part of [2]), a WSN router may receive, transiently (e.g. due to a fortunate environmental reflection), a DIO from another router, much closer to the root – and as a consequence change its parent set and rank to this new more attractive parent. If no stable link exist, this may cause delivery failures.

The Destination Advertisement mechanism, for providing downward routes "from the root to the sensors", remains in

[3]Such is the case for some implementations of IEEE 802.11b. IEEE 802.11b is, of course, not suggested as a viable radio interface for WSNs, but serves to illustrate that such asymmetric designs exist.

a state of flux. While the basic properties of the Destination Advertisement mechanism, given a stable underlying DODAG, appear easy to understand, it does have several inconveniences: all sensor-to-sensor routes transit the root, possibly causing congestion in the wireless spectrum near the root, as well as draining energy from the intermediate routers on an unnecessarily long path. Several solutions are proposed to alleviate this, including allowing intermediate WSN routers, otherwise only forwarding DAO messages towards the root, to record routing state, and allowing these intermediate WSN routers to act as "shortcuts". Another proposed solution is to use proper sensor-to-sensor routing protocols, derived off e.g. AODV [5]. Presently, the DAO mechanism is not fully specified and, thus, difficult to implement and test properly.

Finally, the current specification of RPL does not provide support for "broadcasting" of any form. Unicast traffic to and from the root can be enabled, as previously described, however is inefficient in case the root has data to deliver to all (or a sufficiently large subset) of the WSN routers in the network.

III. Data Broadcasting in RPL

In its current form, RPL does not specify any method for performing data broadcasting through a WSN. This section suggests mechanisms for exploiting the DODAG as constructed by RPL in order to undertake better-than-classic-flooding WSN-wide broadcasting.

The fundamental hypothesis for these mechanisms is similar to that for sensor-to-sensor communication in RPL: all broadcasts are launched from the root of the DODAG. If a sensor needs to undertake a network-wide broadcast, the assumption is that this broadcast is sent to the root using unicast, from where the root will launch the broadcast operation.

A. Parent Flooding (PF)

As a first intuitive optimization over classic flooding, and observing that a broadcast is always launched "at the root", parent-based flooding proposes to restrict a RPL router to retransmit only the first copy[4] of each broadcast packet, received from a "parent". Logically, the basic performance hereof should be similar to that of classic flooding, considering that the broadcast operation in both cases is launched from the DODAG root.

B. Preferred Parent Flooding (PPF)

As a way of getting rid of the "Duplicate Set" for avoiding multiple retransmissions of the same packet, and thus not incur any additional in-router state requirements in WSN routers, preferred parent flooding utilizes the existing relationship between RPL routers to ensure that no router will forward a broadcast packet more than once: as each RPL router is required to select exactly one Preferred Parent, restricting retransmissions of broadcast packets to only those received from the RPL router's preferred parent.

[4]Necessitating the maintenance of a Duplicate Set for detecting which packets have already been retransmitted.

IV. RPL DODAG Performance Study

This section presents results of a simulation study of RPL with the Ns2 simulator. Several properties of the DIO mechanism, as well as unicast and multicast data traffic have been analyzed.

A. Simulation Settings

RPL has been implemented in Java. The specific settings of the scenarios studied are detailed in table I. For each datapoint, the values have been averaged over 10 runs.

Parameter	Value
Ns2 version	2.34
Mobility scenarios	No mobility, random distribution of nodes
Grid size	variable
Node density	50 / km^2
Communication range	250m
Radio propagation model	Two-ray ground
Simulation time	100 secs
Interface type	802.11b
Frequency	2.4 GHz

Table I
NS2 PARAMETERS

1) DIO settings: The implementation reflects a basic version of the RPL protocol: only upward routes, and a single RPL instance with a single DODAG are considered. Since nodes are not mobile in the simulation, the sequence number (and thus the DODAG iteration) will not change during the simulation. At the beginning of the simulation, only the root (which is the node with the ID of 0) starts transmitting DIOs. Nodes other than the root receiving a DIO start sending DIOs exactly two seconds after no more change in their Candidate Neighbor Set has been detected. Each DIO contains the DODAG Configuration suboption.

The simulations have been performed in two variations:

- with periodic DIO transmission: DIOs are sent periodically with an interval of two seconds minus a jitter of maximum 0.5 s (as defined in [6])
- with a trickle timer: `I_min` is 2 s and `I_doublings` is 20. During the simulation, the trickle timer is never reset.

B. Results

This section describes the results of the Ns2 simulation.

Figure 4 shows the maximum and average rank of routers in the DODAG, where the number represents the distance of a node to the root in terms of hops (i.e. the maximum rank represents the diameter of the network, the average rank represents the average over all nodes). The maximum and average ranks grow logarithmically with the number of nodes in the network.

Figure 5 depicts the average number of parents of each node in the DODAG. Keeping the density of the network constant with increasing number of nodes, the average number of parents grows logarithmically.

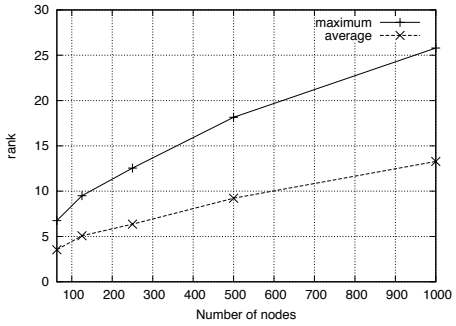

Figure 4. Maximum and average rank of routers in the DODAG

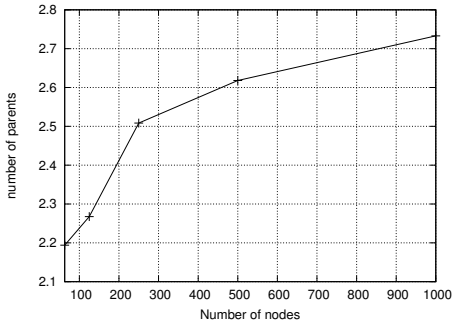

Figure 5. Average number of parents per node in a DODAG

Figure 6 displays the convergence time of the network, i.e. the time that is needed for all nodes that are in the same connected component as the root to join the DODAG. Since each node starts sending DIOs two seconds after the last change to its Candidate Neighbor Set, the convergence time is roughly two seconds times the maximum rank of the DODAG. The convergence time grows logarithmically with the number of nodes in the network.

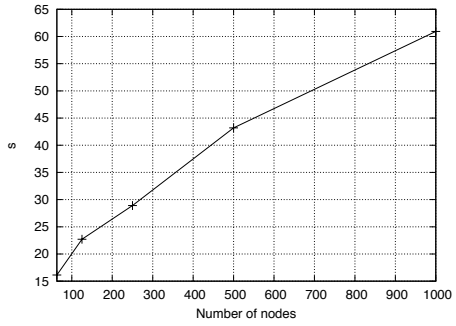

Figure 6. Convergence time

Figure 7 depicts the total control traffic in the network in bytes. The RPL implementation with the trickle timer has significantly less overhead than the periodic timer. The control traffic grows linearly with the number of nodes in the network.

Figure 8 depicts the collision ratio of the DIO messages. Since the RPL implementation using the trickle timer sends significantly fewer DIO messages, the probability of collision is lower.

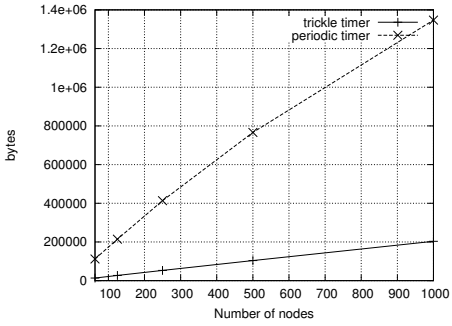

Figure 7. Total control traffic in bytes

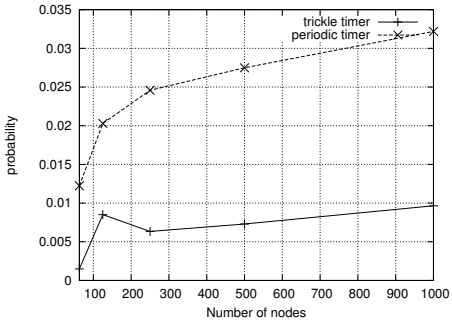

Figure 8. Collision ratio

1) Unicast Data traffic: In the following, unicast CBR data streams of 1280 bytes/s have been sent from an arbitrary node to the root, in average five concurrent streams of 10s duration each.

Figure 9 depicts the delivery ratio of packets that have arrived at the root. It can be seen that it is constantly very high, only few packets are lost due to collisions on lower layers.

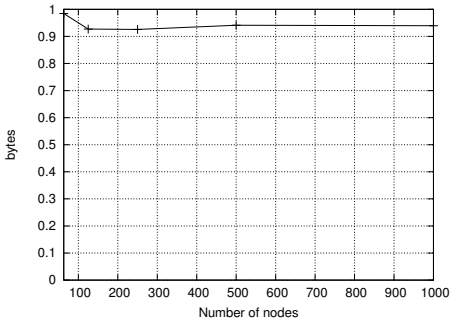

Figure 9. Delivery ratio

Figure 10 illustrates the average path length in number of hops that a data traffic traverses before reaching the root. As expected, it grows logarithmically with the number of nodes, and is very similar to the average rank as depicted in figure 4.

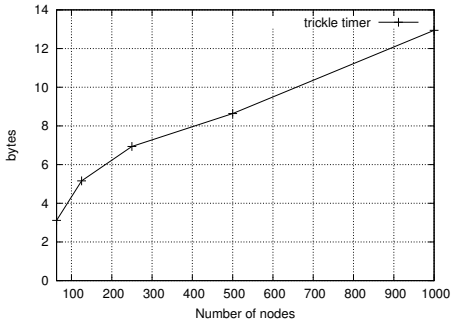

Figure 10. Path length

Figure 11 shows the delay of the data transmission, i.e. the time interval from sending the packet at the source until it reaches the destination. Due to the longer path length, the delay increases with the number of nodes in the network.

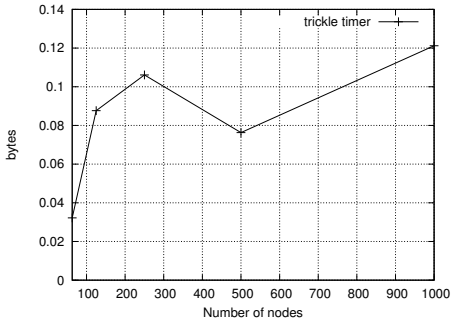

Figure 11. Delay

2) Multicast Data traffic: Figure 12 depicts the delivery ratio of packets that have been broadcasted from the root. Parent Flooding (PF) has a higher delivery ratio than Preferred Parent Flooding (PPF), due to the redundancy of transmissions: when a node receives the same broadcast packet from several of its parents, chances are higher that at least one of the packets will reach the node, while if the one transmission from the preferred parent in PPF is lost due to a collision, the node will not forward the other incoming packets from its (non-preferred) parents. However, PF requires more state than PPF for updating the information about received packets to remove duplicate packets.

Figure 13 illustrates the total retransmission overhead in bytes of broadcasted messages originating at the root, counting each retransmission at every node in the DODAG. The overhead is much lower for PPF than for PF.

V. Conclusion

This paper has presented a critical review of RPL – the currently proposed routing protocol for IPv6-based Wireless Sensor Networks (WSNs), as developed within the Internet Engineering Task Force. A distance vector protocol constructing routing paths from sensors to a central "controller", RPLs basic mechanism is one of DAG formation, with that DAG

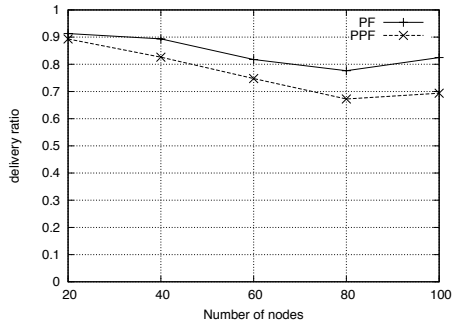

Figure 12. Delivery ratio

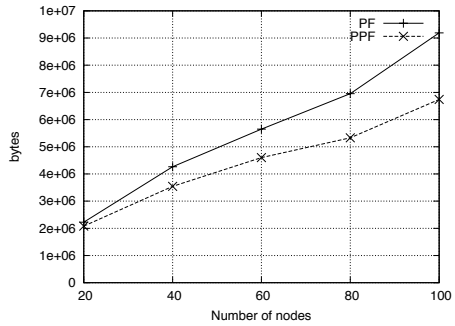

Figure 13. Total retransmission overhead

being the central topology upon which routing is performed. The review reveals areas where, in the authors opinion, further work is required – in particular with respect to tracking of uni-directional links, to point-to-multipoint routes (controller-to-sensor routes) and data broadcasting in a WSN. The paper then suggests a simple zero-in-router-state broadcast protocol, utilizing the DAGs already constructed by RPL.

The paper concludes by a performance study of the "multipoint-to-point" (sensor-to-controller) routing performance RPL, as well as of the suggested data broadcasting mechanisms.

References

[1] J. Martocci et. al., "Building Automation Routing Requirements in Low Power and Lossy Networks", (Work In Progress), http://tools.ietf.org/html/draft-ietf-roll-building-routing-reqs-09, January 2010

[2] The ROLL Design Team, "RPL: Routing Protocol for Low Power and Lossy Networks", (Work In Progress), http://tools.ietf.org/html/draft-ietf-roll-rpl-07, March 2010

[3] P. Lewis, N. Patel, D. Culler, S. Shenker, "Trickle: A Self-Regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks", http://csl.stanford.edu/ pal/pubs/trickle-nsdi04.pdf

[4] T. Narten, E. Nordmark, W. Simpson, H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", Standards Track RFC4861, September 2007

[5] C. Perkins, E. Belding-Royer, S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", Experimental RFC3561, July 2003

[6] T. Clausen, C. Dearlove, B. Adamson, "Jitter Considerations in Mobile Ad Hoc Networks (MANETs)", Informational RFC 5148, February 2008