

# Study of Multipoint-to-Point and Broadcast Traffic Performance in the "IPv6 Routing Protocol for Low Power and Lossy Networks"

Ulrich Herberg, Thomas Heide Clausen

▶ **To cite this version:**

Ulrich Herberg, Thomas Heide Clausen. Study of Multipoint-to-Point and Broadcast Traffic Performance in the "IPv6 Routing Protocol for Low Power and Lossy Networks". Journal of Ambient Intelligence and Humanized Computing, 2011, 2 (4), pp.293-305. 10.1007/s12652-011-0046-2 . hal-02263372

**HAL Id: hal-02263372**

**https://polytechnique.hal.science/hal-02263372**

Submitted on 4 Aug 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Study of Multipoint-to-Point and Broadcast Traffic Performance in the "IPv6 Routing Protocol for Low Power and Lossy Networks"

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

March 4, 2011

**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 "IPv6 Routing Protocol for Low Power and Lossy Networks" (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 several such broadcast mechanisms, all 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.

## 1 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.

### 1.1 WSN Traffic Flows

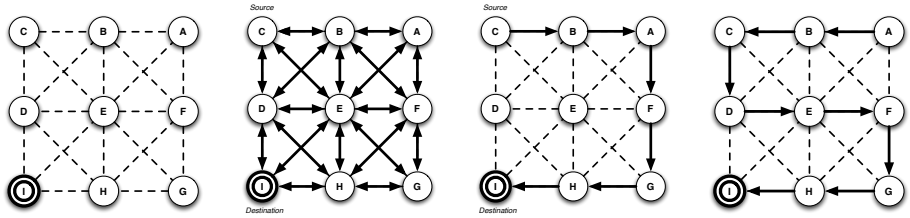
The communications requirements for WSNs are in contrast to "traditional networks", wherein communications devices (network interfaces, switches, routers) have carrying data traffic as their

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].

## 1.2 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.




(a) (b) (c) (d)

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

Consider the example in figure 1(a). The network connectivity is as indicated by the dotted lines, with the source and destination indicated by *Source* (router C) and *Destination* (router I), respectively. A "perceived optimal routing protocol" would be, as illustrated in figure 1(b), 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 1(c) 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 1(d): all routers still retransmitting and receiving as many copies of a packet as in flooding – but with a routing path stretch of 4.

[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-router to sensor-router traffic is assumed a more common occurrence, such as [1].

A flooding operation, as in figure 1(b), would in this case entail 8 transmissions (*i.e.* $(n-1)$ transmissions, with $n$ being the number of routers 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 1(d). On the other hand, the routing path in figure 1(d) 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.

### 1.3 Paper Outline

The remainder of this paper is organized as follows: section 2 provides an overview of the activities of the Internet Engineering Task force (IETF) "Routing over Low Power and Lossy Networks" (ROLL) Working Group, chartered to develop routing protocols for IPv6-based sensor networks, as well as provides a description and critical discussion of the "IPv6 Routing Protocol for Low Power and Lossy Networks" (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 3 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 4 provides a performance study of the multipoint-to-point performance of RPL, as well as a comparative study of the suggested broadcast mechanisms. Section 5 concludes this paper.

## 2 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 in fall 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), a draft version hereof exists [2]. The objective of this protocol is to target networks which *"comprise up to thousands of routers"*, where the majority of the routers have very constrained resources, where the network to a large degree is "managed" by a (single or few)

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

central "superrouters", 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 Directed Acyclic Graph", 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.

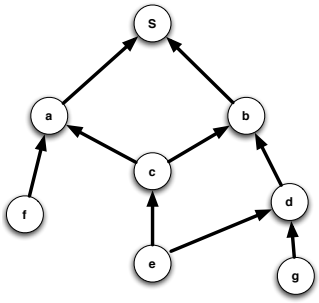



Figure 2: RPL Basic Construct: DODAGs

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.

## 2.1 RPL Data Traffic Flows

The DODAG so constructed is used for installing routes in the WSN routers: the "preferred parent" can serve as a 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 upwards in the DODAG towards the controller, describing which prefixes belong to, and can be reached via, that WSN router. Two distinct "modes of operation" for the downward mechanism are specified in RPL:

(i) In *storing mode*, each router is assumed to maintain routes to all WSN routers in its sub-DODAG, *i.e.* routers that are "deeper down" in the DODAG. DAOs propagate from the routers towards the controller, where each intermediate router adds its reverse routing stack to the DAO message (aggregating routes where possible).

(ii) In *non-storing mode*, only the controller stores routes to all WSN routers in the network. Each WSN router unicasts DAOs to the controller, which then calculates routes to all destinations by "piecing together" the information collected from DAO messages (which contain the destination prefix of the WSN router and addresses of the parents through which it is reachable). In non-storing mode, downward traffic is sent by way of source routing.

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.

## 2.2 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].

## 2.3 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 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.

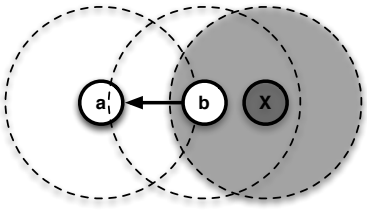



Figure 3: Unidirectional link due to radio interference

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-DODAG, 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. Also, if NUD indicates "no forward progress" based on an upper-layer protocol, there is no guarantee that the problem stems from the preferred parent being unreachable. Indeed, it may be a problem father ahead, possibly outside the WSN, thus changing preferred parent will do nothing to alleviate the situation.

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 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].

Finally, the current specification of RPL does not provide support for "broadcasting" of any

[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.

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.

# 3 Data Broadcasting in RPL

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 that all broadcast operations 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 DODAG root will launch the broadcast operation – this is similar to the basic mechanism for sensor-to-sensor unicast in [2], wherein traffic from the source sensor transits to the DODAG root, for relaying to the destination sensor.

## 3.1 Classic Flooding (CF)

A common baseline for broadcast operations is that of classic flooding: each router relays a broadcast packet upon its first receipt by that router; subsequent receipts of the same packet are suppressed and do not cause retransmissions. This has to its merit that no control traffic is required – however also entails (i) that each data packet must be uniquely identifiable (commonly ensured by embedding a unique sequence number in each broadcast packet, emitted by a given source), (ii) that each router must maintain information (state) for each already received and relayed data packet so as to enable suppression of duplicates, and (iii) each data packet is retransmitted by each router in the network – often with a large degree of redundant transmissions as consequence.

Redundant retransmissions cause increased battery drain, both when transmitting and receiving (and discarding) the redundant packets, and increase contention on the wireless media, increasing the probability of data loss due to collisions. CF is, for these reasons, not suggested as a mechanism for data broadcast in WSNs, but is described here as a baseline for data broadcasting in RPL.

## 3.2 MultiPoint Relay Flooding (MPRF)

A common improvement over Classic Flooding is for each router to select and designate a subset of its neighbors (MultiPoint Relays – MPRs [7]) for relaying broadcast transmissions, thereby reducing the number of redundant retransmissions of each packet. This has been shown to offer dramatic reductions in the network load (fewer transmissions), as well as a dramatic reduction in data loss due to collisions [8].

In order for MPRF to work, a router must select its MPRs such that a message relayed by these MPRs will be received by all routers two hops away, as illustrated in figure 4. To this end, each router must maintain, at a minimum, state describing both its neighbor routers, as well as its 2-hop neighbors ("neighbor routers of neighbors"). MPRF – as CF – requires identification of each broadcast packet, and maintenance of state allowing elimination of duplicate packets.

MPRF is a common approach in wireless ad hoc networks, where it is used *e.g.* for network-wide broadcast of routing protocol control traffic by [9], [10] and [11] – as well as for network-wide data broadcast [12]. Comparing RPL-specific broadcast mechanisms with MPRF is therefore, to a certain extend, a comparison with "state of the art" of broadcasting in wireless multi-hop networks.

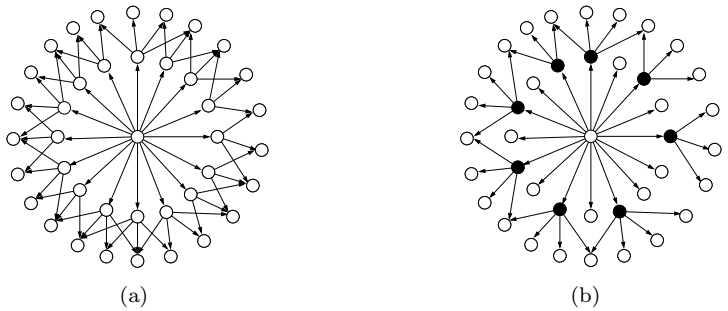



Figure 4: (a) Classic flooding and (b) MPR Flooding

### 3.3 Parent Flooding (PF)

Admitting the RPL "philosophy" of data transmission to sensors originating at (or relaying via) "the DODAG root", RPL lends itself to a first and simple broadcast optimization: restricting a RPL router to retransmit only broadcast packets received from a "parent". Logically, the basic performance hereof should be similar to that of classic flooding: with the broadcast operation initiated from the DODAG root, each router will retransmit the packet upon receipt from a parent. PF does not require any additional control traffic over that which is caused by RPL. PF may apply identification of each broadcast packet, and maintenance of state allowing elimination of duplicate packets in order to avoid multiple retransmissions of the same packet received from different parents – similar to MPRF and CF.

### 3.4 Preferred Parent Flooding (PPF)

In order to not incur any additional in-router state requirements for detecting and suppressing retransmission of duplicate packets, preferred parent flooding utilizes the existing relationship between RPL routers, in order to ensure that no router will forward a broadcast packet more than once. Each RPL router is required to select exactly one Preferred Parent. Restricting retransmissions of broadcast packets to only those received from the router's preferred parent ensures that duplicates received from other routers (parents or otherwise) are ignored for retransmission.

### 3.5 Preferred Parent MPR Flooding (PPMPRF)

PPF is fundamentally a derivative of the MPRF optimization, attempting further to decrease the number of retransmissions necessary for a network wide broadcast. The idea is as follows: each router, selected as "Preferred Parent", must designate a subset of its "selectees" (children which have selected it as preferred parent) as "Preferred Children". These "Preferred Children" must be selected such that a message, relayed by these "Preferred Children", will reach all its "grand children" – *i.e.* the children of its "selectees".

Whenever a router receives a data packet that is to be broadcast throughout the network, that router will only then forward the packet if (i) at least one parent of that router has selected it as preferred child and (ii) the packet has not been previously received (as determined by a duplicated detection mechanism). It is to be noted that it is not sufficient to restrict forwarding to packets received from the preferred parent of a router, but that packets from *any* parent of that router have

to be forwarded if the router has been selected as preferred child by *at least one* of its parents. The rationale is illustrated in figure 5. Assuming that the root of the network (router *0*) has selected *B* as preferred child as indicated by the downward arrow, and *B* forwards a packet originating from the root. If forwarding was restricted to packets received from the preferred parent of a router, *D* would not forward the packet from *B* (since it is no preferred parent of *D*), and thus *X* would never receive the packet.

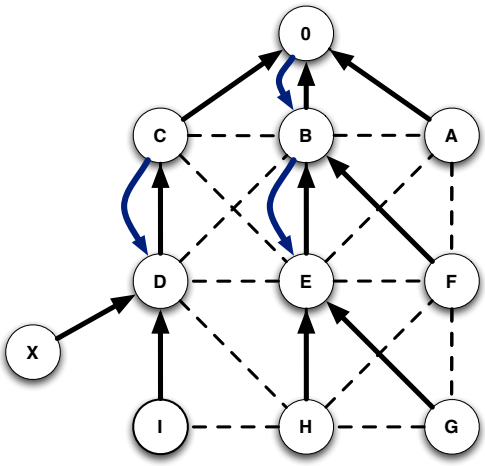



Figure 5: PPMPRF: Example showing the need to forward packets not only received by the preferred parent, but by any parent if the router is selected as preferred child by at least one of its parents. Upward arrows depict preferred parent selection, downward arrows preferred child selection.

Compared to "classic" MPR selection, the "Preferred Children selection" (i) concerns only coverage of "grand children" (*i.e.* "downward" in the DODAG as constructed by RPL) and (ii) is restricted by the preferred parent selection from RPL.

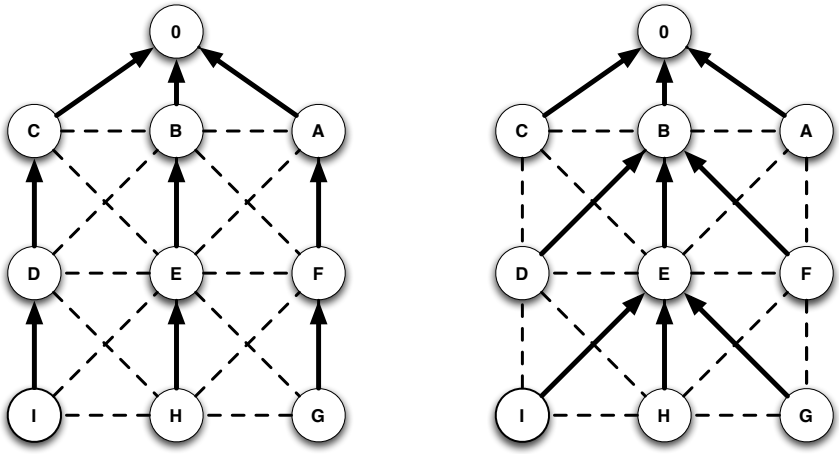
This restriction entails less liberties with respect to selecting relays for "best 2-hop coverage". It is quite possible that the child providing the "best" coverage of a router has not selected that router as Preferred Parent, and that therefore PPMPRF will result in more relays than MPRF. In RPL, the Preferred Parent selection is intended to optimize for "best upwards paths towards the DODAG root" (possibly according to some deployment specific optimization criteria), which may not coincide with what would be optimal for "best downwards coverage".

The PPMPRF mechanism also requires that each router knows (i) which children have selected it as Preferred Parent (*i.e.* its *selectees*), and (ii) which routers are Preferred Children of these selectees. This information can be made available through adding an option to DIO messages, emitted by all routers running RPL.

### 3.6 Optimized Preferred Parent MPR Flooding (PPMPRF-opt)

This mechanism represents a small optimization over PPMPRF, in that it provides all neighboring routers with the same rank with information, encouraging coordinated Preferred Parent selection so as to try to reduce the number of routers selected as Preferred Parent. Thus, a router will select as

its Preferred Parent among its parents, the one which most of its adjacent routers also have in their parent set. Given a tie, the parent which a majority of the adjacent routers have already selected as Preferred Parent will be chosen. Thus, in addition to the information indicated for PPMPRF, PPMPRF-opt requires all parents to be advertised.




(a) (b)

Figure 6: Uncoordinated PP selection (a) and coordinated PP selection (b) in the same network. Solid arrows indicate the selection of a Parent as Preferred Parent; dotted lines the connectivity of the network.

Figure 6(a) depicts an example of Preferred Parent selection, as may happen in basic RPL: a router selects its Preferred Parent amongst all its parents with the lowest rank in an uncoordinated way. Worst case (in terms of redundant transmissions and therefore possible collisions when broadcasting), routers D, E and F all select different Preferred Parents (C, B, and A respectively). Similarly, I, H, and G may select three different Preferred Parents. For PPMPRF, this means that all routers, other than 0, will be selected as MPRs and thus retransmit a broadcast.

Figure 6(b) depicts a coordinated Preferred Parent selection. Router D will advertise all its parents (C and B) in its control messages, as will E (parents C, B and A) and F (parents B and A). D has an equal choice between parents C and B, and F has the same choice between B and A. E will select B as Preferred Parent because this is the only parent that both of its adjacent routers can also select as Preferred Parent. Once D and F receive a control message from E, advertising that B is selected as Preferred Parent, they will also select B. Thus, only routers B, E and H will be selected as Preferred Parents and therefore retransmit a broadcast.

Such coordinated Preferred Parent selection may be a double-edged sword for RPL. While it is a potential benefit for broadcast traffic from the DODAG root, unicast traffic flows towards the DODAG root via Preferred Parents. Thus, coordinated selection of Preferred Parents implies that unicast traffic is concentrated through a subset of the routers in the network, possibly increasing congestion in these routers, increasing the battery drain in these routers etc.

# 4 RPL 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 broadcast data traffic have been analyzed.

## 4.1 Simulation Settings

RPL has been implemented in Java. The specific settings of the scenarios studied are detailed in table 1. For each datapoint, the values have been averaged over 10 runs.

Table 1: Ns2 parameters

| Parameter | Value |
|---|---|
| Ns2 version | 2.34 |
| Mobility scenarios | No mobility, random distribution of routers |
| Grid size | variable |
| Router 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 |

### 4.1.1 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 routers 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 router with the ID of 0) starts transmitting DIOs. routers 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 message option.

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 `DIOIntervalDoublings` is 20. During the simulation, the trickle timer is never reset.

## 4.2 Results

This section describes the results of the Ns2 simulation. Figure 7 shows a RPL instance of a simulated network with 1000 routers. In this depicted setting, and in all simulated scenarios in the following, routers are not mobile and randomly distributed in the square simulation area.

Figure 8 shows the maximum and average rank of routers in the DODAG, where the number represents the distance of a router to the root in terms of hops (*i.e.* the maximum rank represents the

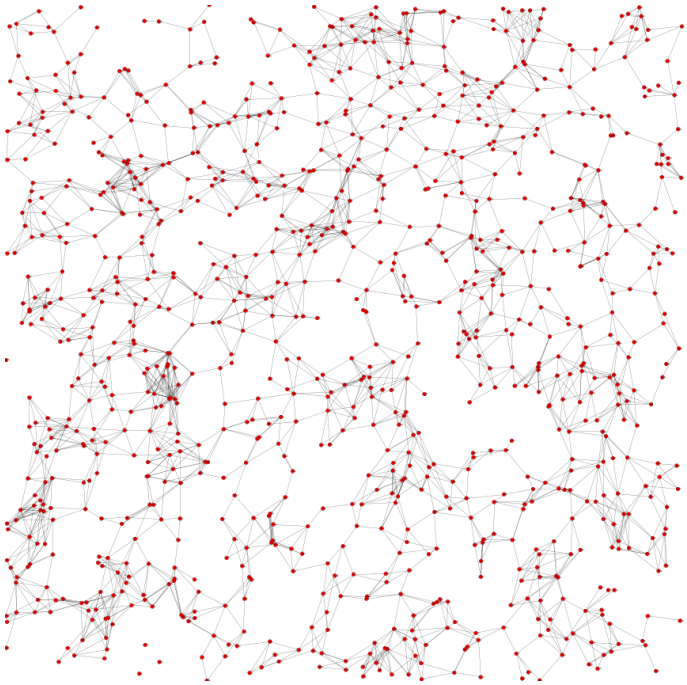


Figure 7: Example RPL instance with 1000 routers

diameter of the network, the average rank represents the average over all routers). The maximum and average ranks grow logarithmically with the number of routers in the network.

Figure 9 depicts the average number of parents of each router in the DODAG. Keeping the density of the network constant with increasing number of routers, the average number of parents grows logarithmically.

Figure 10 displays the convergence time of the network, *i.e.* the time that is needed for all routers that are in the same connected component as the root to join the DODAG. Since each router 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 routers in the network.

Figure 11 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 routers in the network.

Figure 12 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.

#### 4.2.1 Unicast Data traffic

In the following, unicast CBR (Constant Bit Rate) data streams of 1280 bytes/s have been sent from an arbitrary router to the root, in average five concurrent streams of 10s duration each.

Figure 13 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.

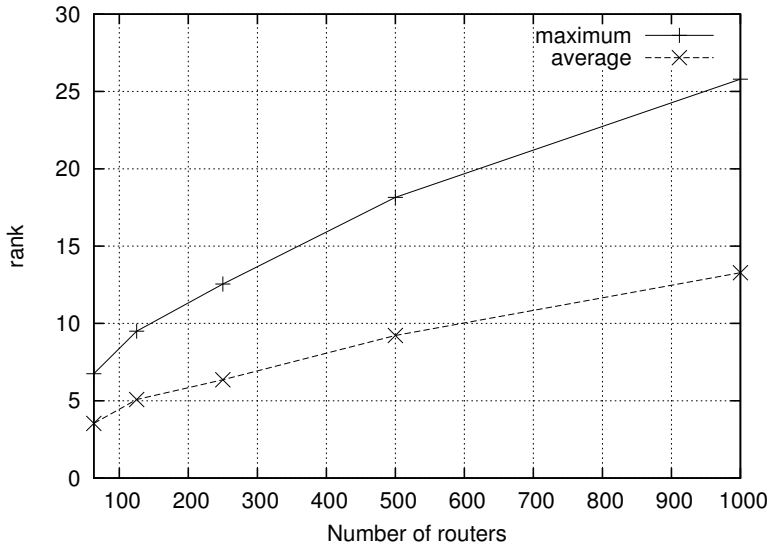



Figure 8: Maximum and average rank of routers in the DODAG

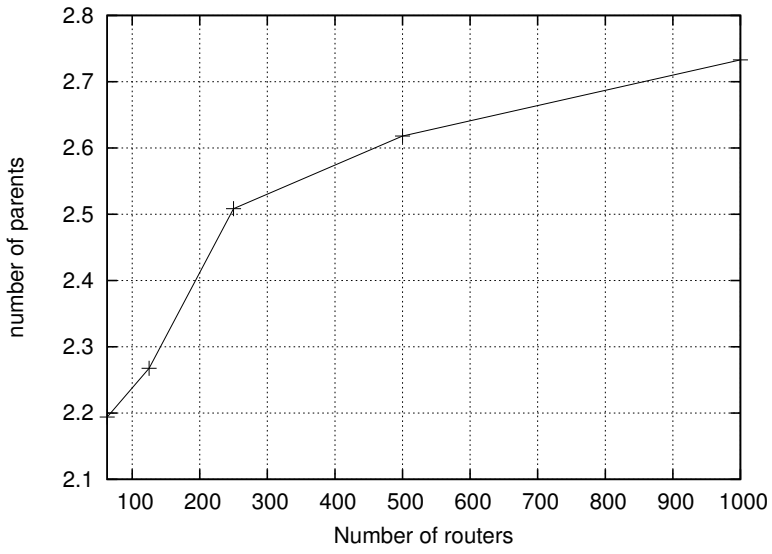



Figure 9: Average number of parents per router in a DODAG

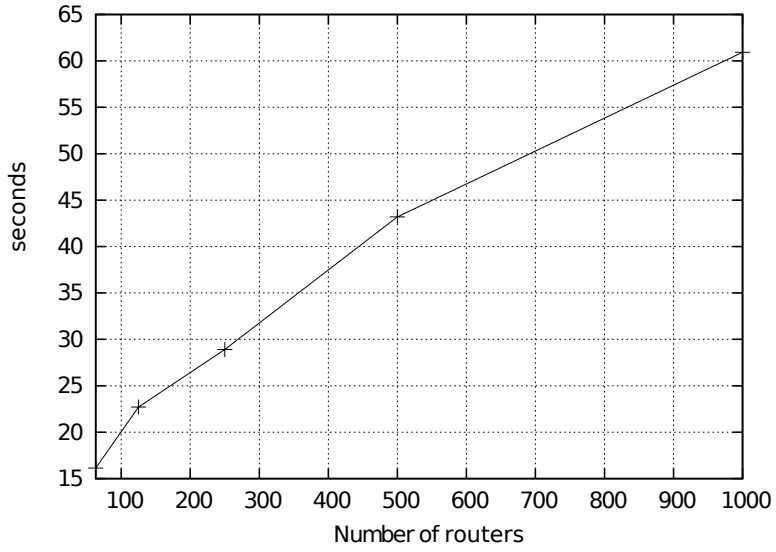



Figure 10: Network convergence time

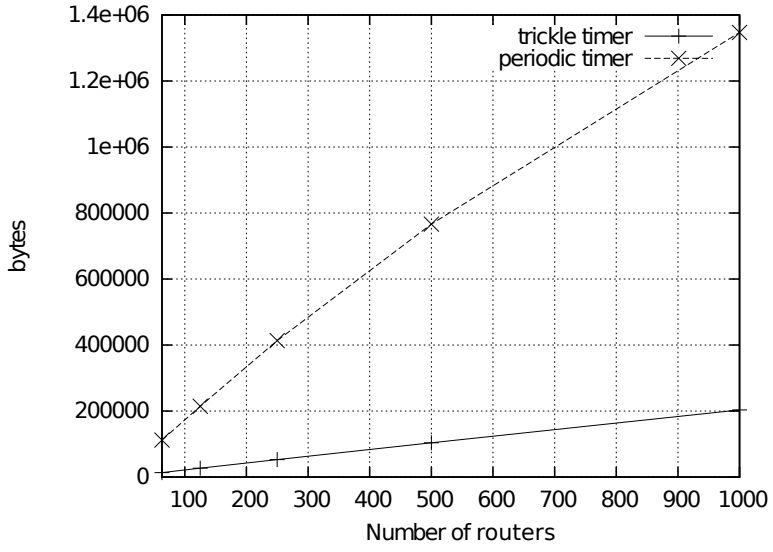



Figure 11: Control traffic: overhead in bytes

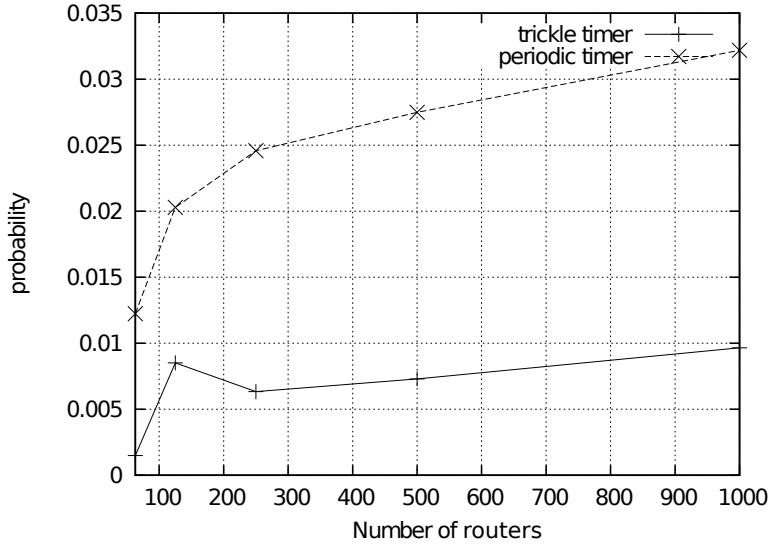



Figure 12: Control traffic: collision ratio

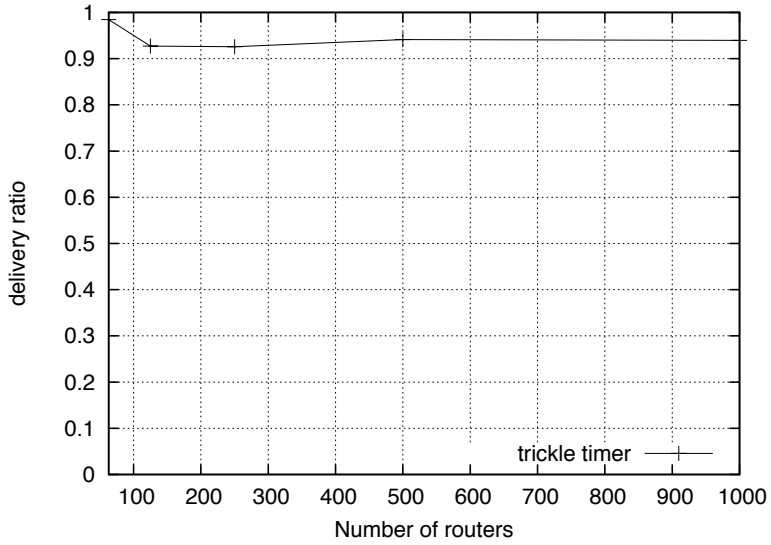



Figure 13: Unicast: delivery ratio

Figure 14 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 routers, and is very similar to the average rank as depicted in figure 8.

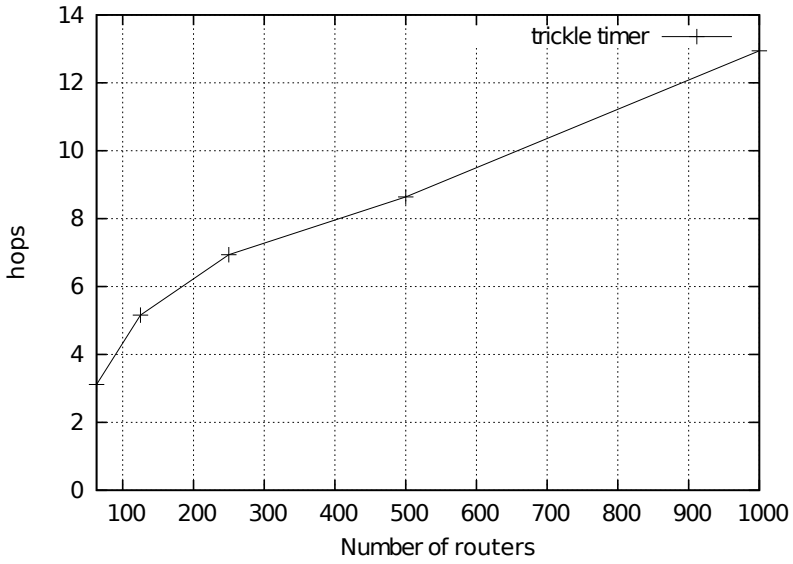



Figure 14: Unicast: path length

Figure 15 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 routers in the network.

#### 4.2.2 Broadcast Data traffic

In the following, the broadcast mechanisms presented in section 3 are analyzed in terms of MAC layer collisions, delivery ratio, overhead, delay, and path length. CF and PF (without duplicate detection) are not considered since their performance is expectedly much worse than any of the other mechanisms.

Figure 16 depicts the number of collisions of frames on the MAC layer, for the different broadcast mechanisms. MPRF and PPMPRF-opt yield the lowest number of collisions among the analyzed protocols, with PPMPRF causing about the same number of collisions as PF+DD (PF with duplicate packet detection). This is expected, as in MPRF, relays are explicitly selected so as to avoid redundant retransmission by topologically close routers, and the coordinated preferred parent selection in PPMPRF-opt also reduces the number of relays. PPMPRF without coordination entails more relays, as more routers in the network will be selected as preferred parents, which in turn select the relays (*i.e.* preferred children). In PPF, topologically close routers are likely to have chosen the same Preferred Parent and so will explicitly produce redundant retransmissions. Consider the example in figure 17, wherein a broadcast transmission is made by router 0 and relayed as indicated by the solid arrows. In PPF, as indicated in figure 17(a), each router will select its Preferred Parent and retransmit the packet once upon receipt from that preferred parent. Routers

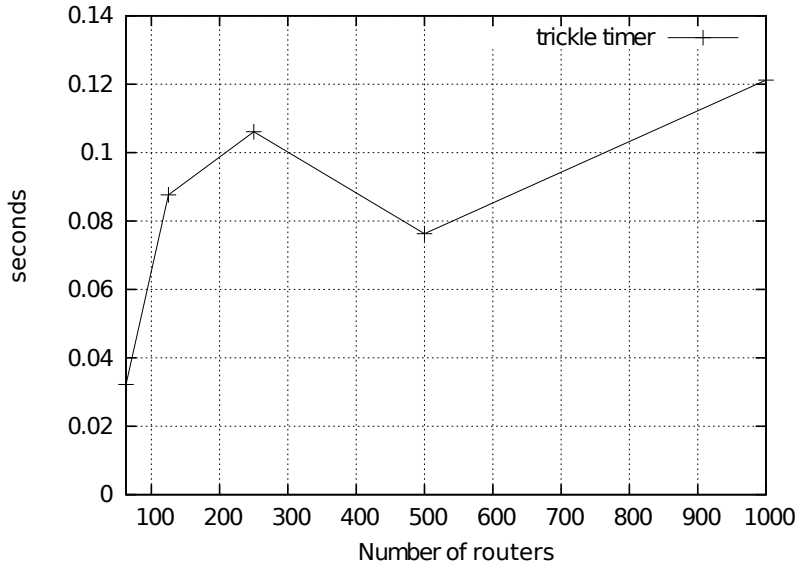



Figure 15: Unicast: delay

A, B and C all receive the transmission directly from router 0. Routers D, E and F have all chosen one of A, B and C as Preferred Parent and will thus all retransmit when receiving the transmission from their chosen preferred parent – similar for I, H, G, even though these three do not have any routers further down the network. In contrast, in figure 17(b), MPRs have been selected. Router 0 has selected B as MPR (as B "covers" D, E, F) and router B has chosen router E as MPR (as it covers all of G, H, I). As there are no further routers "below" in the network, router E has chosen no MPRs downwards. Thus, only B and E retransmit the broadcast packet from 0 – *i.e.* for each "level" in this simple network, only a single transmission occurs, with no collisions at each level.

Figure 18 depicts the delivery ratio of broadcast packets. The delivery ratio of the MPRF and PPMPRF mechanisms are the highest of the compared broadcast mechanisms, with PPMPRF-opt being not much below MPR. This can be interpreted as a tradeoff between redundancy and efficiency: in relatively scarce networks (such as the simulated scenario) a higher redundancy of relays, such as in PPMPRF, can lead to a higher delivery ratio, despite of the increased number of collisions. as observed in figure 16. In dense networks, however, the large number of collisions with more redundant delays can reverse that effect and reduce the delivery ratio. A detailed analysis of the MPR relaying mechanism can be found in [7].

PF+DD has a higher delivery ratio than PPF, due to the redundancy of transmissions – when a router receives the same broadcast packet from several of its parents, chances are higher that at least one of the packets will reach the router, while if the one transmission from the preferred parent in PPF is lost due to a collision, the router will not forward the other incoming packets from its (non-preferred) parents. The higher delivery ratio of PF+DD is at the expense of vastly higher media load, as depicted in figure 19: the cumulative number of bytes transmitted during the simulations are significantly higher for PF+DD and for PPMPRF without the optimization. PPF incurs a lower overhead than PF+DD with MPRF still outperforming PPF by a large, and constant, margin. PPMPRF-opt has a similar overhead as MPRF.

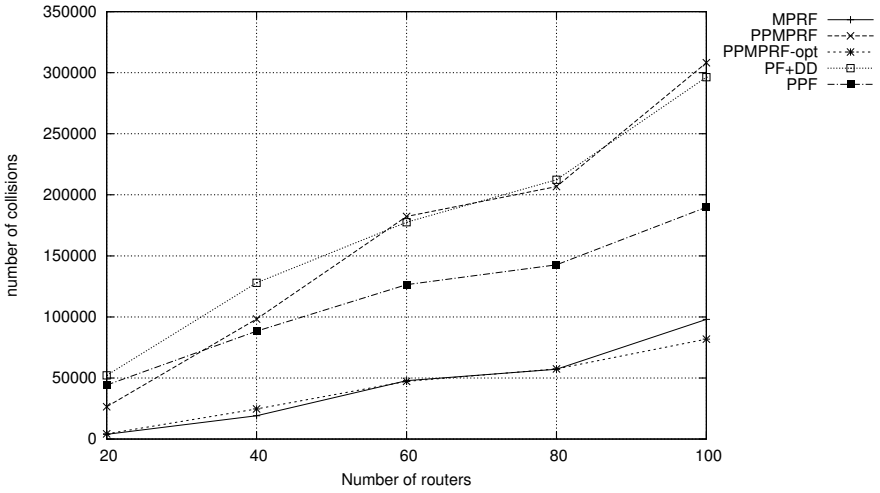
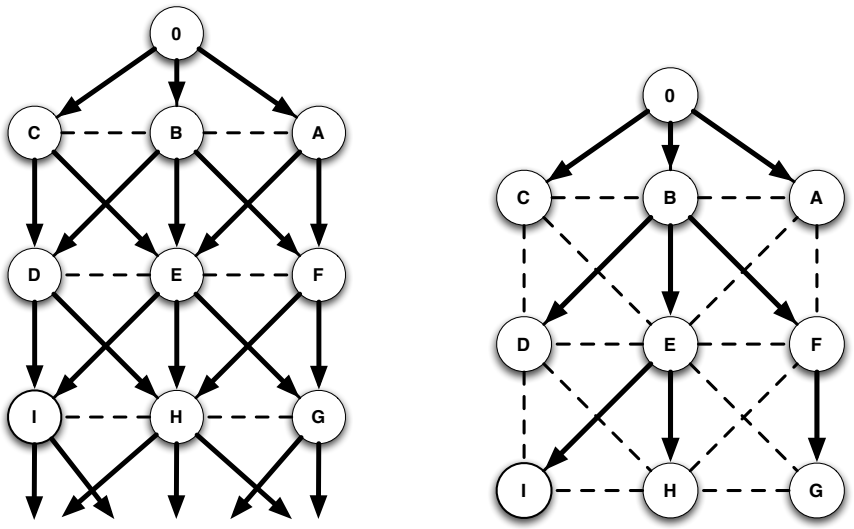



Figure 16: Broadcast: total number of MAC layer collisions




(a) PPF, originated by router 0

(b) MPRF, originated by router 0

Figure 17: PPF (a) and MPRF (b) in the same network. Solid arrows indicate transmission of a packet; dotted lines the connectivity of the network.

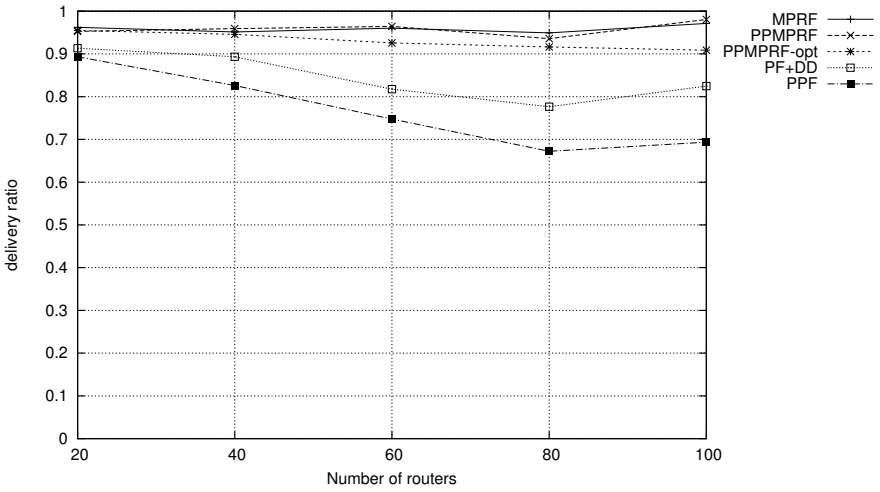



Figure 18: Broadcast: delivery ratio

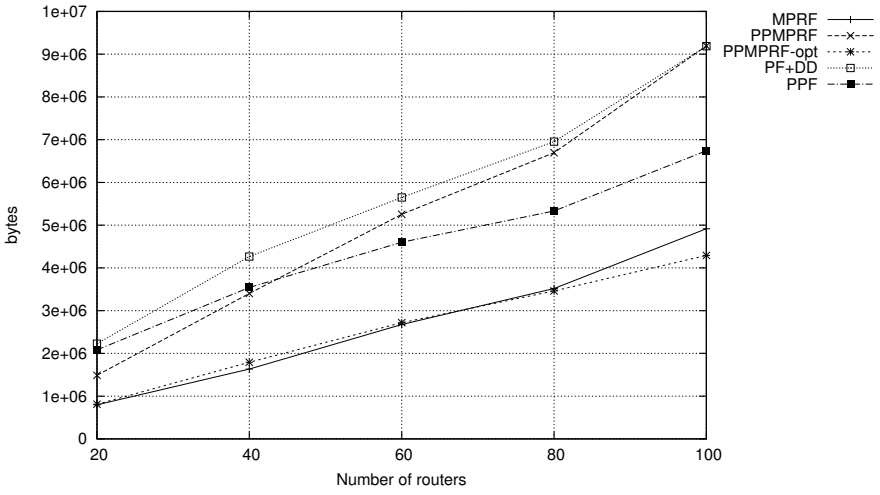



Figure 19: Broadcast: total retransmission overhead

Figure 20 depicts the average end-to-end delay for data traffic from the root to every WSN router in the network, and figure 21 depicts the average path length of successfully delivered data packets. The optimized MPR-based broadcast mechanisms incur the lowest delay of the protocols, while PPF causes a slightly lower delay than does PF+DD. The, on average, longer path lengths of MPRF are due to the data delivery ratio being higher – MPRF successfully "reaches" routers farther away from the root (as depicted in figure 22). It has been shown ([7]) that MPR leads to optimal path length. That means that every mechanism indicating a shorter path in the figure entails a lower reachability of routers further away from the broadcast source. Longer paths indicate suboptimal paths. It is worth observing that MPRF achieves the optimal path length with a lower delay still. This can in part be explained by the fact that MPRF ensures that data is flooded via shortest paths, and in part by the fact that with fewer retransmissions, less media and queue contention occurs.

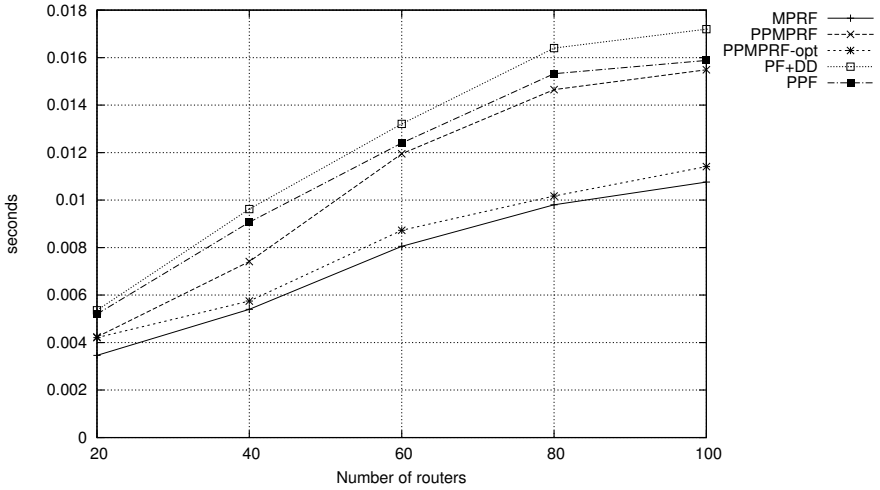



Figure 20: Broadcast: average delay

## 4.3 PPF with Jitter

In the results presented in section 4.2.2, data traffic has been promptly forwarded by each WSN router, without explicit delay. As has been shown in [**?**, 6], adding a random jitter before retransmitting a broadcast packet can significantly reduce the number of collisions and, therefore, increase the delivery ratio for broadcast packets. In the following, the effect of adding jitter to PPF is investigated.

Figure 23 depicts the collision ratio of frames when using no jitter, and a random jitter uniformly distributed between 0 and 500 ms respectively. With jitter, the collision ratio is much lower than it is without. This is due to the fewer concurrent retransmissions by adjacent WSN routers. Comparing to figure 16, PPF with jitter yields a collision ratio comparable to, or lower than, MPRF without jitter.

As a consequence of the lower collision ratio, the delivery ratio of PPF with jitter is higher than it is without, as depicted in figure 24. Comparing to figure 18, the delivery ratio of PPF still remains consistently below that of MPRF, even when PFF is used with jitter.

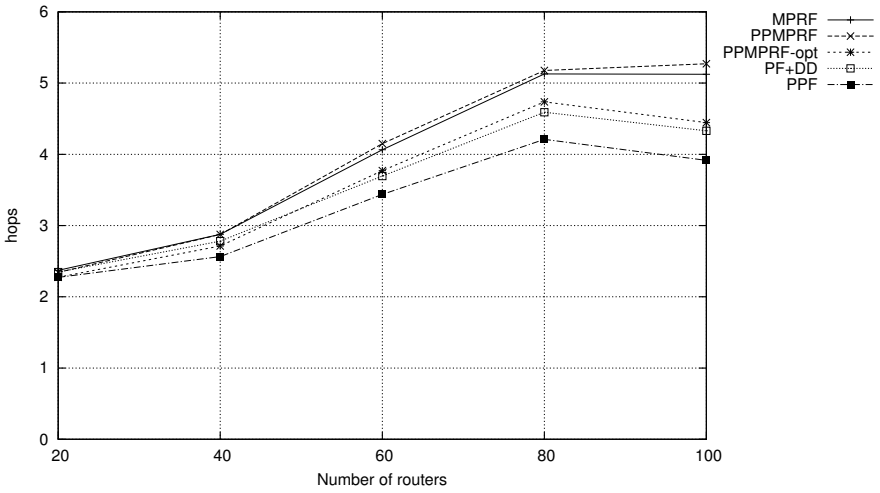



Figure 21: Broadcast: average path length

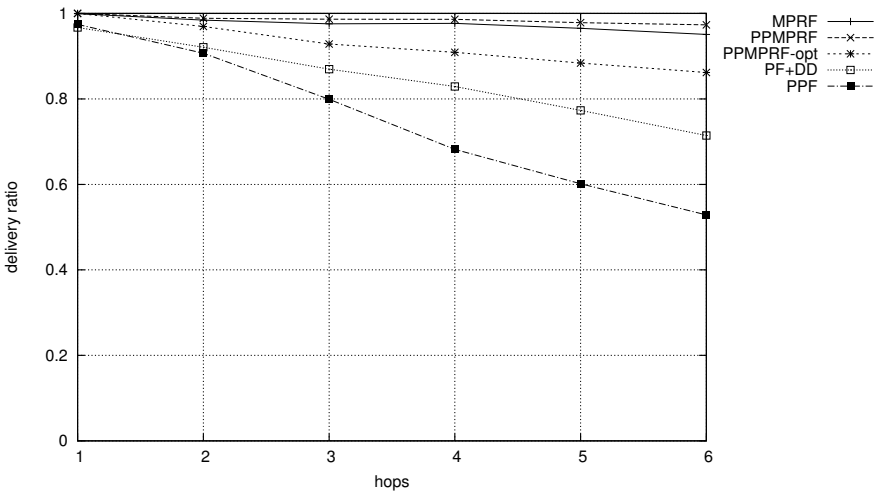



Figure 22: Broadcast: traffic delivery ratio with respect to distance from the root in hops (with 100 routers in the network)

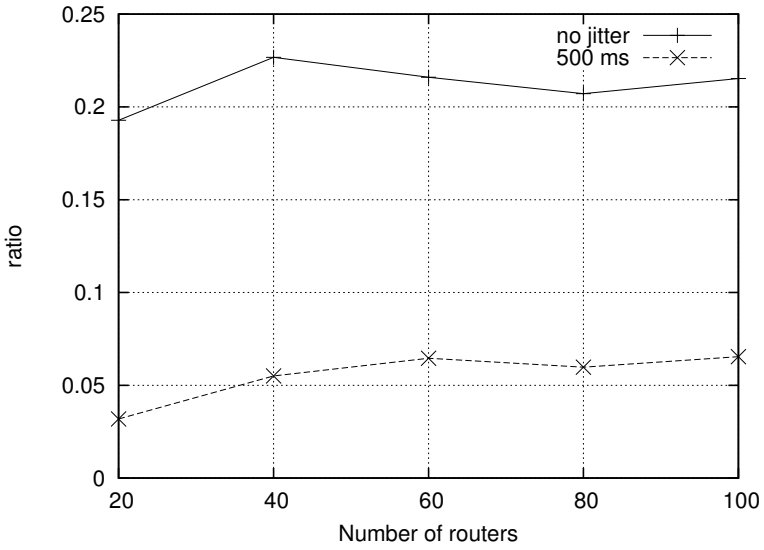



Figure 23: Collision ratio of PPF with jitter

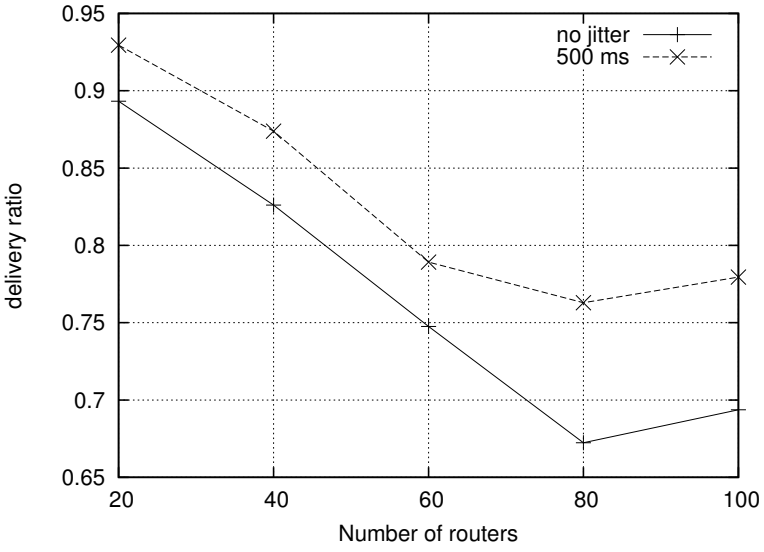



Figure 24: Delivery ratio of PPF with jitter

The drawback of using jitter is a higher end-to-end delay of packets, as depicted in figure 25. With jitter, the delay is considerably higher than it is without.

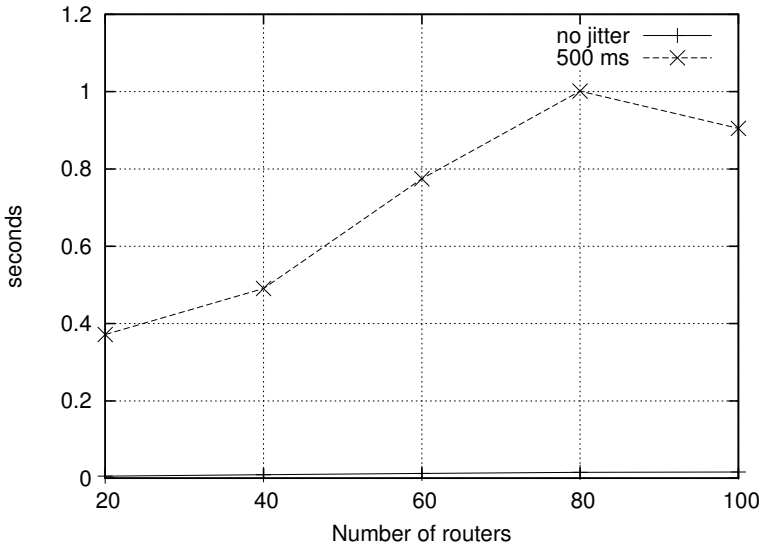



Figure 25: Average delay of PPF with jitter

# 5 Conclusion

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

Moreover, the performance of "upward" traffic (sensor-to-controller) in an RPL-based sensor network is evaluated using Ns2 simulations, showing that RPL scales well with the size of the network. The control traffic overhead of the protocol is relatively low, and delivery ratio of data traffic is high and stable – even for large networks of several hundred routers.

The paper then presents a comparative study and an Ns2-based performance evaluation of broadcast mechanisms for RPL, utilizing the DAGs already constructed by RPL. These broadcast mechanisms, denoted "Parent Flooding" (PF), "Preferred Parent Flooding" (PPF) and "Preferred Parent MPR Flooding" (PPMPRF) adhere to the "root-oriented" concept of RPL, in that all broadcast operations are to be initiated by the root of the DAG. PF, PPF and PPMPRF are studied and compared by way of network simulations – and as a point of comparison, MPR Flooding (MPRF), is also subjected to the same network scenarios in the simulator.

The results show that the MPR based broadcast mechanisms lead to a much lower bandwidth

consumption and a higher delivery ratio than PF and PPF, with the expense of additional control traffic information that is required. If such information is not provided, PPF leads to a better performance than PF. In order to further reduce collisions in the network, using a jitter (*i.e.* a delay when forwarding a message on a router) is evaluated: the jitter decreases the collisions, and thus increases the delivery ratio, but also increases the end-to-end delay.

# References


[1] J. Martocci, P. De Mil, N. Riou, W. Vermeylen, "Building Automation Routing Requirements in Low-Power and Lossy Networks", Informational RFC5867, http://tools.ietf.org/html/rfc5867, June 2010

[2] T. Winter, P. Thubert, RPL Author Team, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks", (work in progress), http://tools.ietf.org/html/draft-ietf-roll-rpl-18, February 2011

[3] P. Lewis, N. Patel, D. Culler, S. Shenker, "Trickle: A Self-Regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks", Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation (NSDI), 2004

[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

[7] A. Qayyum, L. Viennot, A. Laouiti, "Multipoint relaying: An efficient technique for flooding in mobile wireless networks", 35th Annual Hawaii International Conference on System Sciences (HICSS), 2001

[8] T. Clausen, L. Viennot, T. Olesen, N. Larsen, "Investigating Broadcast Performance in Mobile Ad-hoc Networks", Proceedings of the IEEE conference on Wireless Personal Multimedia Communications (WPMC), October 2002

[9] T. Clausen, P. Jacquet, "Optimized Link State Routing Protocol (OLSR)", Experimental RFC3626, http://www.ietf.org/rfc/rfc3626.txt, October 2003

[10] T. Clausen, P. Jacquet, D. Nguyen, E. Baccelli, "SPF Multipoint Relay (MPR) Extension for Ad Hoc Networks", Experimental RFC5449, http://www.ietf.org/rfc/rfc5449.txt, February 2009

[11] T. Clausen, C. Dearlove, P. Jaquet, "The Optimized Link State Routing Protocol version 2 (OLSRv2)", http://tools.ietf.org/html/draft-ietf-manet-olsrv2-11 (work in progress), April 2010

[12] J. Macker, "Simplified Multicast Forwarding", http://tools.ietf.org/id/draft-ietf-manet-smf-10 (work In progress), March 2010