Ad Hoc Networks 54 (2017) 69–84

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

BMRF: Bidirectional Multicast RPL Forwarding

Guillermo Gastón Lorente[a], Bart Lemmens[a], Matthias Carlier[a,b,*], An Braeken[b], Kris Steenhaut[a,b]

[a] Department of Electronics and Informatics (ETRO), Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussel, Belgium
[b] Department of Industrial Engineering (INDI), Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussel, Belgium

ARTICLE INFO

Article history:
Received 15 March 2016
Revised 10 September 2016
Accepted 5 October 2016
Available online 6 October 2016

Keywords:
Wireless Sensor Networks
Multicast
RPL

ABSTRACT

Nowadays, the transition of Wireless Sensor Networks (WSNs) to Internet Protocol version 6 (IPv6), in particular to IPv6 over Low power Wireless Personal Area Networks (6LoWPAN), is evident [1]. However, in most commonly used implementations, not all IPv6 features are available. For example, current implementations are not very optimized for multicast, despite the many benefits multicast can offer with respect to the number of radio transmissions and the amount of consumed energy.

In this paper we present Bidirectional Multicast RPL Forwarding (BMRF), a new multicast protocol that combines the best features of the Routing Protocol for Low Power and Lossy Networks (RPL) multicast on the one hand and of Stateless Multicast RPL Forwarding (SMRF) on the other hand. The main features are bidirectionality and the ability to offer a choice between Link Layer broadcast and Link Layer unicast for which the threshold to decide for a mote, which link layer mode to choose, is mainly based on its number of interested children and the duty cycling rate. An implementation of BMRF is realized in Contiki. Our measurements show that BMRF, when using the optimal configuration, results in less radio transmissions, and less energy consumption, and higher packet delivery ratio compared to SMRF, often at the cost of a higher end-to-end delay.

1. Introduction

1.1. Multicast

In computer networks, multicast (point-to-multipoint or multipoint-to-multipoint distribution) is a group communication technique in which information is sent to a set of destination machines using the notion of multicast address [2].

IP multicast is an implementation for point-to-multipoint communication over an IP-based network infrastructure. The destination nodes send, join and leave messages (to subscribe or to unsubscribe from the information flow). IP multicast uses the network infrastructure efficiently, by allowing the source to send a packet only once, even if it needs to be delivered to a large number of destinations. The routers in the network only duplicate the packet if it is required to reach multiple nodes.

1.2. Multicast in Wireless Sensor Networks

Network protocols and applications for WSNs have special requirements due to the constrained resources available in sensor nodes. Those constrained sensor nodes are also called motes. The short range and huge exposure to noise and interference of the radio communication, the small battery power supply and limited memory and processing capacity restrict the networking capabilities of those motes. Therefore, WSNs need dedicated protocols, designed with these issues and limitations in mind, in particular the energy efficiency concern.

Here, 6LoWPAN [3] and RPL [4] enter the scene. They allow the use of IPv6 inside resource constrained WSNs. This explains the importance of these protocols for the deployment of the Internet of Things (IoT). In 6LoWPAN, the physical protocol used in the wireless link layer is usually the physical layer of IEEE 802.15.4 [3]. In the link layer there is also a Carrier Sense Multiple Access (CSMA) based Medium Access Control (MAC) protocol and often also a Radio Duty Cycling (RDC) protocol (for example: ContikiMAC) to put the radio to sleep when inactive (see Section 2.3). A typical WSN protocol stack is shown in Fig. 1.

* Corresponding author.
E-mail address: matcarli@vub.ac.be (M. Carlier).

http://dx.doi.org/10.1016/j.adhoc.2016.10.004

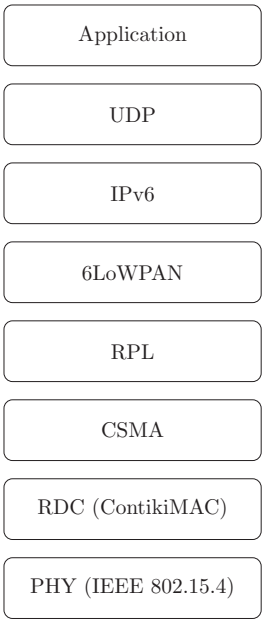

Fig. 1. Typical protocol stack for a WSN mote.

Multicast is one of the communication techniques that IPv6 provides, which looks promising for WSNs, since it could allow to reduce memory usage and radio transmissions, improving the overall energy efficiency.

As an example, we focus on a mote that needs to send measurements periodically to a group of devices. Instead of maintaining a list of interested devices and sending a unicast packet per subscribed mote, it could send a single packet carrying a multicast destination address. Doing so, duplicate information transmission is avoided and memory consumption is reduced since no list of subscribers needs to be kept. This example occurs in practice when a temperature sensor has to inform many other devices (such as the heating control, an interactive wall display or a smartphone about its measurement). Another case in which multicast may be useful is the transmission of a command to several actuators. Having a multicast address to reach those actuators, avoids keeping a list of their IP addresses in the mote. A multicast packet can be sent to that group and all actuators will get the command. Other real world examples are: turning on all the lights of a room, changing the refreshment timer parameter of all temperature sensors in a house or turning off all air conditioning machines when no presence in the house is detected and negotiating a group key for ensuring secure and authenticated communication [5]. A multicast group is called dynamic if motes can subscribe to it and unsubscribe from it at any moment. This notion can also avoid many unnecessary transmissions. A practical example showing the benefit of dynamic multicast groups could be the heating control. During the day it needs information from the temperature sensors inside the house, but at night it is programmed to turn itself off. If it unsubscribes from the multicast group, unnecessary messages will be prevented from being transmitted. In the morning, when it turns on, it could resubscribe, reactivating the multicast flow.

The ability to communicate with groups of resources is important for many IoT applications in general and for building automation systems in particular. Therefore CoAP, which is expected to play an important role as an application protocol, for use in constrained environments (to replace the role of HTTP) has also been extended with group communication capabilities [6,7]. The authors of [8] compare the two underlying communication type choices being unicast or multicast IPv6 for organizing group communication for COAP and demonstrate that it comes to a trade-off between reliability and speed. As IPv6 multicast protocol they choose the existing Stateless Multicast RPL Forwarding (SMRF) protocol (see Section 2.3) in a sensor network with RDC switched off. Also service discovery is an application that can benefit from IPv6 multicasting [9]. Authors of [9] propose a lightweight multicast forwarding for service discovery in low-power IoT networks. IEEE 802.15.4 Link Layer (LL) protocol does not natively support multicast. Therefore, Link Layer broadcast or Link Layer Unicast is used to deliver the frames. SMRF is an example of an IPv6 multicast protocol based on RPL that uses Link Layer Broadcast, whereas Wang [10] proposes one that uses Link Layer unicast with node mobility as extra feature.

Our work will focus on improving/broadening the scope of the existing IPv6 (6LoWPAN) multicast protocols, by inter alia making the right choices at the underlying link layer to propagate the multicast packet i.e. via Link Layer unicast(s) or Link Layer broadcast.

1.3. Contribution and paper structure

This paper presents a new multicast protocol, called Bidirectional Multicast RPL Forwarding (BMRF). BMRF is implemented and compared with the existing Stateless Multicast RPL Forwarding (SMRF) protocol, showing some interesting results.

SMRF is a LL broadcast based RPL driven multicast protocol, most efficient in WSNs with RDC switched off. It aims at minimizing the amount of sent packets and delivery delay and has small memory usage, just keeping in each node the IPv6 multicast address for which the mote has interested children.

RPL multicast is a LL unicast based RPL driven multicast protocol, most efficient in WSNs with RDC protocol enabled (such as ContikiMac and XMAC), especially in situations with relatively low duty cycling rate or/and few interested children. In RDC-enabled WSNs, it also refrains the number of sent packets, offers good reliability and takes care about the delay with a reasonable extra memory space needed for keeping the list of interested children associated to an IPv6 multicast address in each node.

This paper proposes BMRF, able to offer a choice between LL broadcast and LL unicast for which the decision for each mote is based on a threshold mainly determined by the mote's number of interested children and the channel check rate. When the motes always choose BMRF LL unicast (independent of the threshold), then BMRF is a well designed and well functioning implementation of RPL unicast with some extra features. When the motes always choose BMRF LL broadcast, then BMRF behaves like SMRF but adds some nice extensions, but has a slightly higher memory consumption than SMRF.

Section 2 discusses the operation, benefits and shortcomings of the most important existing multicast protocols in literature. In Section 3, we explain the features and operation of the proposed BMRF protocol. A detailed evaluation study is performed in Section 4 and conclusions are given in Section 5.

2. Related work

Multicast issues of IPv6-based WSNs have already been addressed by other researchers. In this section we describe how these issues have been tackled and discuss general operation, benefits and drawbacks of the proposed solutions.

2.1. MPL

Multicast Protocol for Low power and Lossy Networks (MPL) [11], also often referred to as Trickle multicast, was an initial attempt of the IETF Routing Over Low power and Lossy networks (ROLL) working group to address IPv6 multicast forwarding in constrained networks.

MPL provides multicast communication through a controlled network-wide flooding governed by Trickle timers [12]. Every mote in the network receives all multicast packets, regardless of whether it has subscribed to the multicast group or not. Each mote disseminates the multicast packets to all of its neighbors.

Due to the use of a network-wide flood in which all motes broadcast every multicast packet, a mechanism to suppress duplicates is required. By including a sequence number with an MPL option in the IPv6 header, multicast packets can be distinguished. Multicast packets must also be buffered to detect whether a specific packet has previously been received.

Trickle timers are used to schedule the periodic retransmission of the buffered packets in order to disseminate them through the network. Periodic control packets are exchanged between neighbors to communicate the state of their respective buffers. When it is detected that a particular mote did not received a given packet, that packet can be delivered by one of the mote's neighbors. The periodicity of the control packets is governed by Trickle timers.

MPL's simplicity offers several advantages:

- Thanks to its flooding, MPL does not use a distribution tree and as such does not require any routing protocol nor topology maintenance.
- Multicast group registration is not needed since all motes receive every multicast packet. As such, there is no (un)subscription overhead.
- With appropriate parameter values for the Trickle timer configuration, MPL can offer high packet delivery ratios (PDRs). Since all packets are buffered by all motes, recovering from a failed transmission is trivial. Considering that links in WSNs suffer from high loss rates and instability, MPL can cope well with packet loss.

However, MPL also suffers from some drawbacks:

- Since the multicast packets are disseminated throughout the whole network, a huge communication overhead is introduced if the number of motes interested in the multicast traffic is small. Moreover, the periodic retransmission of both control and data packets increases the communication overhead even further. A major incentive however to use multicast in WSNs is to achieve better transmission efficiency (besides the ease for the programmer to refer with one send command to many devices).
- The packet buffer requires a lot of memory. Since motes have very limited memory, the number of packets that can be stored is small, which can reduce MPL's efficiency.
- Multiple Trickle timers, one per buffered packet and an additional one for the control packets, are required which consume both memory and processing power.
- Delivery disorder avoidance is not available, which means out-of-order delivery of the multicast packets can happen. However, packets can be reordered based on the sequence number, taking into account an additional processing overhead. This problem only affects applications for which the ordering of packets matters.

2.2. RPL

Routing Protocol for Low power and lossy networks (RPL) [4] is an IPv6 routing protocol designed by the IETF ROLL working group to address the specific needs and constraints of WSNs. It constructs a Destination-Oriented Directed Acyclic Graph (DODAG), which defines parent-child relationships between the different motes. It supports point-to-point (between motes inside the network), point-to-multipoint (multicast), and multipoint-to-point (towards a central control point) traffic flows.

A single mote is designated as the root of the DODAG. This root mote is often a more powerful node, which acts as gateway to the Internet. Motes exchange control packets and gradually select their parents, and a single preferred parent from them, creating a tree topology, called RPL tree. Fig. 2 shows a RPL tree topology as example.

RPL distinguishes two different kinds of routes:

- Downward routes: from a mote to any of its successors. For example, in Fig. 2 from mote 6 to mote 36.
- Upward routes: from a mote to any of its ancestors. For example, in Fig. 2 from mote 28 to mote 1.

RPL also defines four Modes of Operation (MOP):

- **MOP 0**: No Downward routes maintained by RPL. This mode only supports multipoint-to-point traffic for which motes can send packets to the RPL tree's root.
- **MOP 1**: Non-Storing Mode of Operation. Downward routes are supported but all IPv6 packets should be forwarded to the root which maintains all downward routes. A special IPv6 header option [13] is used to source-route the packets from the root of the tree.
- **MOP 2**: Storing Mode of Operation with no multicast support. The individual motes support downward routes by maintaining a routing table for their successors.
- **MOP 3**: Storing Mode of Operation with multicast support. Identical to the previous MOP with the additional support for point-to-multipoint traffic flows.

The remainder of this section focusses on MOP 3, since it is the only mode with multicast support, and explains the forwarding and group registration mechanisms.

Destination Advertisement Object (DAO) packets are used for multicast group registration. These packets are identical to unicast DAO packets, except from the RPL Target option in the DAO, which indicates the multicast group of interest. The RPL RFC specifies whether a router can send its multicast DAO to only its preferred parent or to a set of its parents. In the first case, if a transmission would fail on the link to the preferred parent, motes in the subtree below that parent will not receive any multicast packet. For the second case, multiple paths will provide redundancy but also cause duplicates. The RPL RFC does not discuss how to detect and discard these duplicates.

The forwarding mechanism essentially reduces the problem to multiple unicast transmissions. When a router receives a multicast IPv6 packet, it forwards the packet through a Link Layer unicast transmission to all its children in the DODAG which have previously expressed their interest for that specific multicast group with a multicast DAO. If the packet originated from within the tree, the routers also have to forward the packet to their preferred parent until the RPL root is reached. Otherwise, potential subscribers in the group of children of the root (and their children), cannot be reached. The forwarding mechanism is illustrated in Fig. 3.

In Fig. 3a, mote 22 is the source of the multicast packet and motes 17 and 28 are subscribed to the multicast group. As shown in the left image of Fig. 3b, mote 22 sends the multicast packet to its preferred parent and to all its children that are interested in the multicast group. The right image of Fig. 3b demonstrates how all routers propagate the packet towards the RPL tree's root.

RPL offers several benefits for multicast traffic compared to MPL:

- RPL uses the same tree topology for unicast and multicast traffic. Multicast packets are only disseminated to those parts of the network in which motes have registered to the multicast group, which improves transmission efficiency.

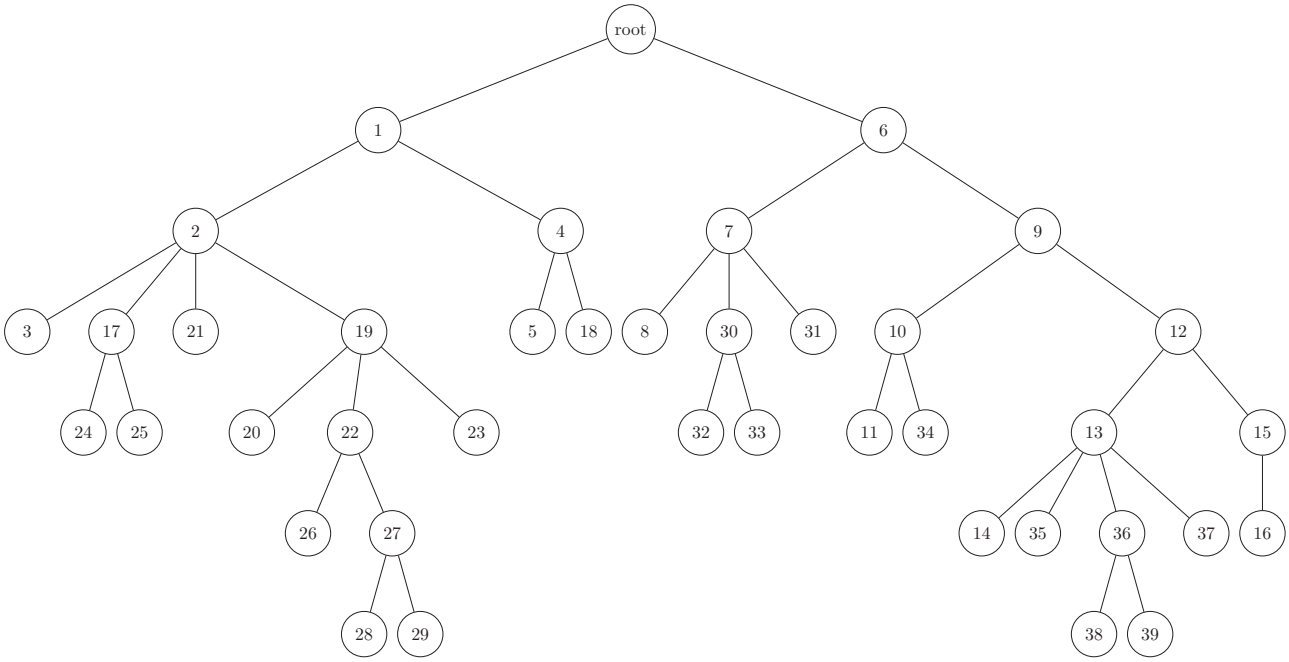

Fig. 2. Example of RPL tree, representing a DODAG in which only the connections of a mote with its preferred parent have been kept.

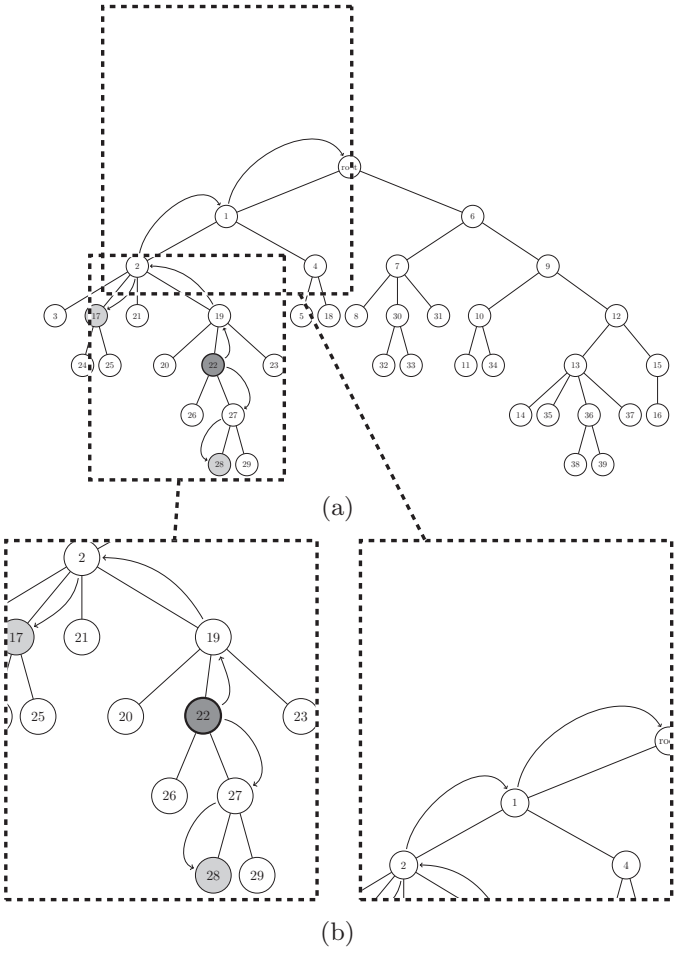

Fig. 3. Multicast forwarding mechanism in RPL MOP 3.

- Group registration happens through DAO packets, identical to unicast DAOs, so no additional control packets need to be defined.
- By using the standard RPL forwarding mechanism, RPL multicast makes sure all packets are delivered in the order they are sent.

There are however still some issues:

- RPL multicast does not explicitly do duplicate avoidance. If a mote subscribes to a multicast group via multiple parents, each of these parents may deliver a copy of the same multicast packet.
- When a router has to forward a multicast packet to its interested children, it uses a Link Layer unicast transmission per child. Since every unicast transmission consumes energy and takes a certain amount of time, it might be more appropriate to use a Link Layer broadcast transmission to deliver the multicast packet to all interested children at once.

2.3. SMRF

Stateless Multicast RPL Forwarding (SMRF) [14,15] tries to address the shortcomings of RPL multicast. SMRF specifies a new forwarding mechanism for RPL MOP 3 and retains the multicast group management technique described in the RPL RFC.

To overcome the duplicates problem, SMRF dictates that motes should only process multicast packets received from their preferred parent. SMRF does not specify how multicast packets can travel upwards the RPL tree, restricting the source of any multicast traffic to be the root of the tree.

SMRF uses Link Layer broadcasts to increase the transmission efficiency when forwarding a multicast packet to the interested children. Since the Link Layer protocols for WSNs often use Radio Duty Cycling (RDC), special attention regarding broadcast efficiency is required. To better understand these cross-layer effects, we briefly illustrate how ContikiMAC [16], a popular RDC protocol, works.

Most RDC protocols periodically turn on the radio transceiver to listen for incoming packets. The duration of this period is

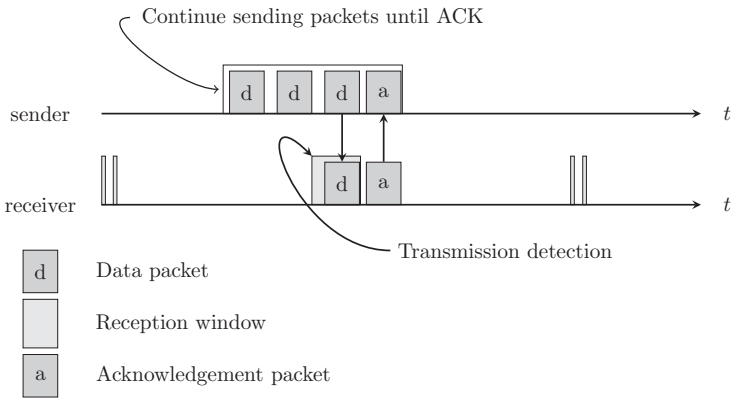

Fig. 4. ContikiMAC unicast transmission.

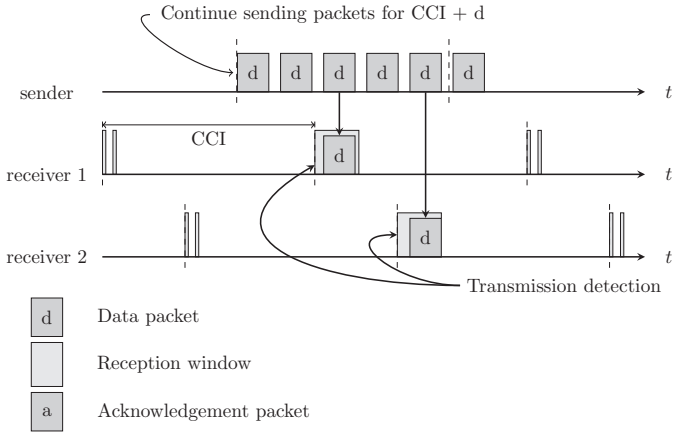

Fig. 5. ContikiMAC broadcast transmission.

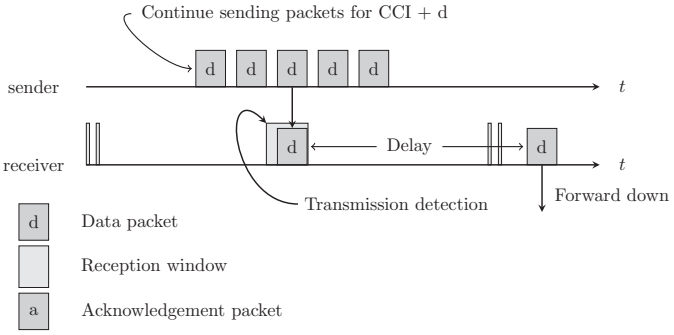

Fig. 6. SMRF delay operation.

referred to as the Channel Check Interval (CCI). The transmitter consecutively sends the packet multiple times to assure that one of the packets finds a woken up receiver. The transmitter can use a phase lock to remember the offset within the CCI at which a receiver previously woke up, in order to minimize the number of transmissions when it decides to send another packet. If a mote detects traffic, it keeps its radio awake until the complete packet has been received.

For unicast transmissions, the receiver will respond with an acknowledgement which allows the sender to stop transmitting and the sender can also benefit from the phase lock, as shown in Fig. 4. In broadcast mode, the sender is obliged to transmit for at least one CCI period to assure that all of its neighbors have woken up once. This mechanism is illustrated in Fig. 5. Notice that Link Layer acknowledgements are not possible for ContikiMAC broadcasts.

When receiving a packet via Link Layer broadcast, a mote cannot immediately forward that packet since it would collide with the ongoing broadcast transmissions by the sender. Therefore, SMRF introduces a short delay before further transmitting the received multicast packet, as shown in Fig. 6.

As mentioned above, SMRF solves some of RPL multicast weaknesses:

- SMRF avoids duplicates by processing only multicast packets received from the preferred parent.
- Link Layer broadcast is used instead of multiple Link Layer unicasts to try to reduce energy consumption.

SMRF does suffer from a couple of drawbacks as well:

- Only downwards forwarding is possible, which limits the source of the multicast traffic to be the root of the RPL tree.
- The additional delay required to prevent collisions increases end-to-end latency.

Moreover, whether a Link Layer unicast or broadcast is more efficient in terms of energy depends on the number of receivers and the CCI (see Section 3.2.3: Mixed mode). When the number of addressed receivers is small, multiple Link Layer unicasts might yield a lower total number of transmissions, since the receivers can notify the sender to stop transmitting by acknowledging the packet and the sender can benefit from the phase-lock. Thus, when a mote has only a single child interested to the multicast group, that mote should preferably use a Link Layer unicast to forward the multicast packet to the child.

2.4. ESMRF

Enhanced Stateless Multicast RPL Forwarding (ESMRF) [17] adds an extension to SMRF that allows sources of multicast traffic to be located within the network. Whenever a mote wants to send a multicast packet, it encapsulates that packet in an Internet Control Message Protocol version 6 (ICMPv6) delegation packet which is sent to the root of the tree. The root then distributes the multicast packet down in the network on behalf of the source.

ESMRF has the same characteristics as SMRF, except that it also supports multicast sources besides the root of the tree. However, all multicast traffic is routed via the root. This can become expensive if multicast sources and their associated group members are in each other's vicinity. Our solution, presented next, avoids this waste.

3. Bidirectional Multicast RPL Forwarding (BMRF)

In this section we describe the characteristics and operation of our proposed multicast forwarding mechanism, called Bidirectional Multicast RPL Forwarding (BMRF).

3.1. BMRF features

The following desired functionalities were taken into account while developing BMRF:

- Configurable forwarding. We have argued that the choice for Link Layer unicast or broadcast to forward a multicast packet should depend on the number of interested children. BMRF can operate in three modes: unicast, broadcast, and mixed mode. Unicast mode behaves the same as RPL multicast, whereas broadcast mode acts similarly to SMRF. When using mixed mode, either a Link Layer unicast or broadcast is used depending on whether the number of interested children is larger than a configurable threshold.
- Bidirectionality. Multicast packets are forwarded both up and down in the RPL tree. This essentially means the same as allowing multicast sources within the network.
- Duplicates avoidance. This problem was solved by SMRF. BMRF also avoids duplicates.
- Delivery disorder avoidance. This characteristic is directly inherited from RPL. Unless another network layer modifies the packet order, BMRF will deliver multicast packets in the correct order.
- Multi-sourcing. More than one mote can send multicast packets to the same multicast destination address. This is not mentioned in the RPL RFC but it should work without any modifications. Since SMRF can only send multicast packets to the entire tree from the root, multi-sourcing is not applicable to SMRF.
- Dynamic group registration. The RPL RFC does not mention anything about unsubscribing from a multicast group. However, a DAO with a multicast group address as Target value and a lifetime equal to 0 will act as an unsubscribe packet. When such a DAO is sent to the preferred parent, that parent removes the route corresponding to that group for the child. If there are no other children interested in that group, the parent itself unsubscribes from the multicast group.

3.2. BMRF operation

BMRF introduces a new forwarding mechanism compared to SMRF, but keeps the same multicast group management technique as SMRF, which is in line with the RPL RFC (MOP 3). In the following sections we describe how BMRF operates in a mote that is the source of a multicast packet and in a mote that has to forward a multicast packet. We introduce the different Link Layer modes that BMRF can use for forwarding. If unicast is used for downward forwarding then BMRF is one of the possible implementations of RPL multicast. Fundamentally, BMRF is an extensive revision of the SMRF algorithm, adding support for multiple new features, such as upwards forwarding, group unsubscriptions and overall better RPL No-Path DAO handling and the ability to forward downstream by using multiple link-layer unicasts, instead of a single broadcast. All these features will be explained in detail in the next paragraphs.

3.2.1. Source mote wants to send a multicast packet

When a source mote wants to send a multicast packet, it has to send that packet both up and down in the RPL tree. Exceptions to this rule are when the mote is the root (so there are no motes up in the RPL tree) or if there are no interested children (so there is no need to send the packet down in the tree).

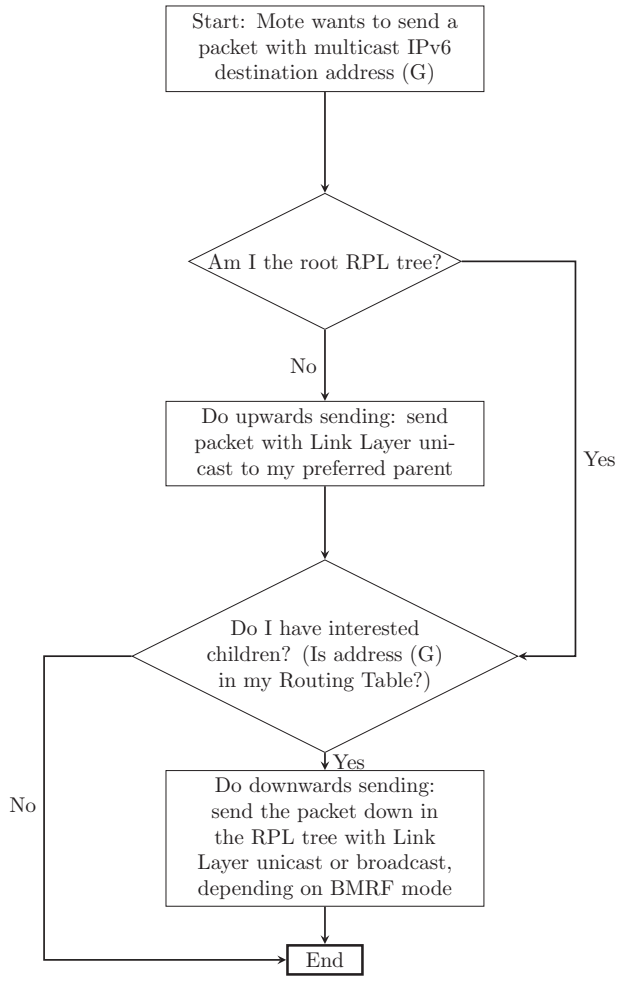

Fig. 7. Sending a multicast packet with BMRF.

This sending algorithm is illustrated in the flowchart in Fig. 7. In the flowchart, one can see that the mote checks whether it is the root of the tree and if not, it sends the packet throught a Link Layer unicast to its preferred parent.

Next, it checks whether it has interested children and if so, the packet is sent downwards through Link Layer unicast or broadcast, depending on the configured mode (see Section 3.2.3). To check whether there are interested children, the routing table is checked for entries, containing the multicast address.

Why link layer unicast for upwards sending?. It is reasonable to question why a Link Layer unicast must be used to send the packet to the preferred parent, when the multicast packet is broadcasted to the interested children using Link Layer broadcast anyway. One might assume that this Link Layer broadcast could be exploited to send the packet upwards in the tree as well, in order to avoid the additional unicast to the preferred parent. However, this mechanism can result in multiple parents receiving the packet, which will all forward the packet. This will produce duplicates. Therefore, it is desired that the packet is only processed by the source's preferred parent. As the parent cannot detect if a child has selected it as a preferred parent, Link Layer unicast is the only option to send the packet upwards and avoid duplicates. Although a mote can distinguish motes above and below it in the tree, using the routing table or the RPL rank included in the RPL Hop-by-Hop option header (HBHO) [18] in the IPv6 header, it cannot know if a mote below it, has selected it as its preferred parent since the child never com-

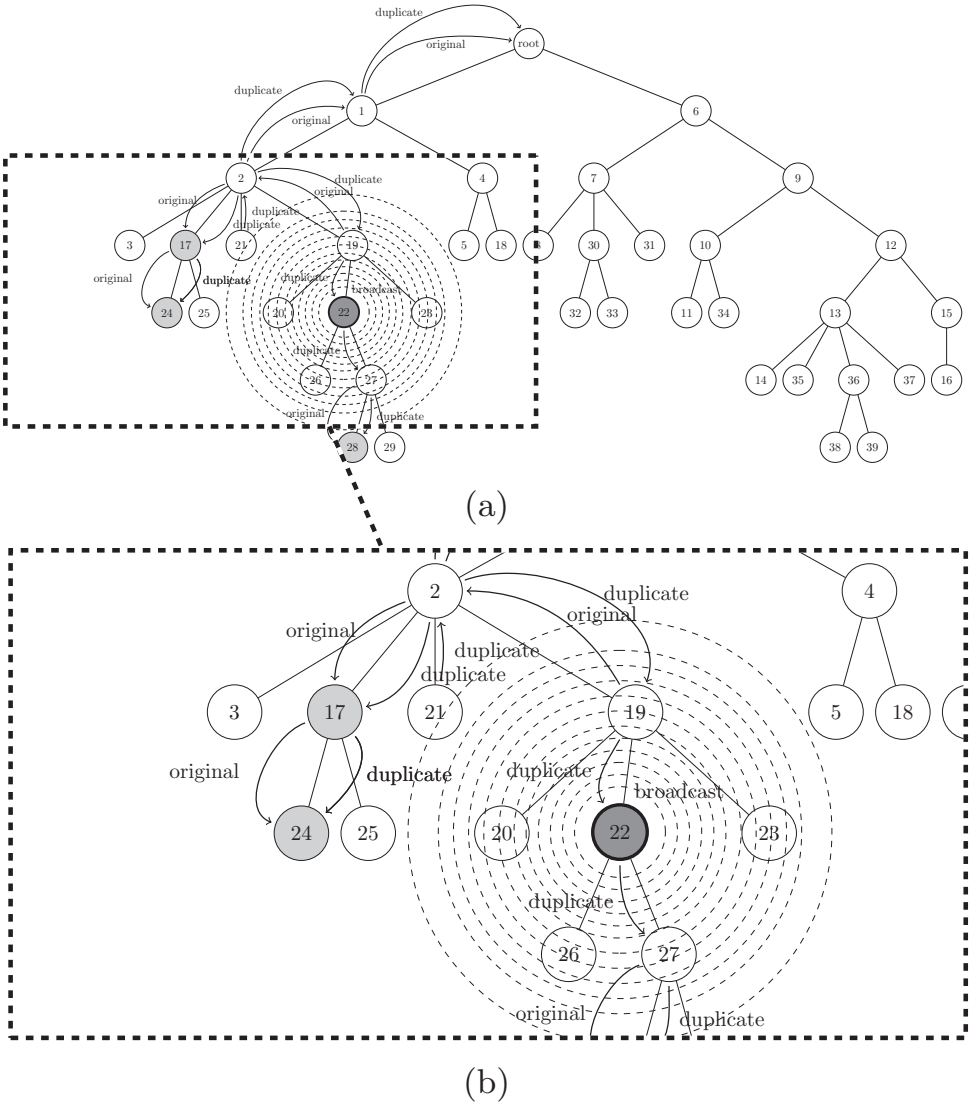

Fig. 8. Duplicates when sending up with Link Layer broadcast in BMRF.

municates this information. Indeed, an explicitly addressed unicast must be used to send the packet upwards.

Fig. 8 illustrates the previous explanation with a tangible example. Mote 22 is the source of the multicast packet. It uses a Link Layer broadcast to send out the multicast packet. Amongst others, mote 27 (a router that has an interested child) and mote 19 (the preferred parent) receive the multicast packet. Mote 27 will forward it downwards to its interested children (mote 28) and mote 19 will forward it upwards to its preferred parent (mote 2). Mote 2 will forward it up and down again (amongst others to mote 1 and mote 17).

Until now, no transmission created duplicates, and the transmission efficiency is better than using the additional upwards unicast, since mote 22 has delivered the packet with a single transmission. The problem is that mote 21 also receives the Link Layer broadcast transmission from mote 22. The only check it can perform before accepting the packet is to look whether it comes from above or from below. Since mote22ʳs rank shows that it is below mote 21 in the RPL tree, the packet will be accepted, and thus creates a duplicate.

If mote 21 would be able to know that it is not mote 22's preferred parent, it could discard the packet, avoiding the duplicate. Since this is not possible, Link Layer unicast must be used to send the packet upwards.

3.2.2. Mote forwards a multicast packet

The forwarding algorithm of BMRF is more complex compared to the one proposed by SMRF. The forwarding algorithm is illustrated in the flowchart shown in Fig. 9.

- **If a packet is received from above**, it will only be accepted if its origin is the preferred parent, which can be checked via the Link Layer source address. If accepted, the routing table is consulted. If any entry is found for the multicast group, the packet is forwarded downwards. If the packet has been received via a Link Layer broadcast, a short delay is introduced before forwarding to avoid collisions, identical to SMRF's delay mechanism. Whether a Link Layer unicast or broadcast frame is used depends on BMRF's selected mode (see 3.2.3).
  The mote also checks whether itself is a member of the multicast group. If it is, the packet is pushed upwards in the network stack.
- **If the packet is received from below** with a MAC unicast address, this means it has been sent by a child which has selected this mote as its preferred parent. Only then, the mote processes the packet and consults its routing table. If any entry

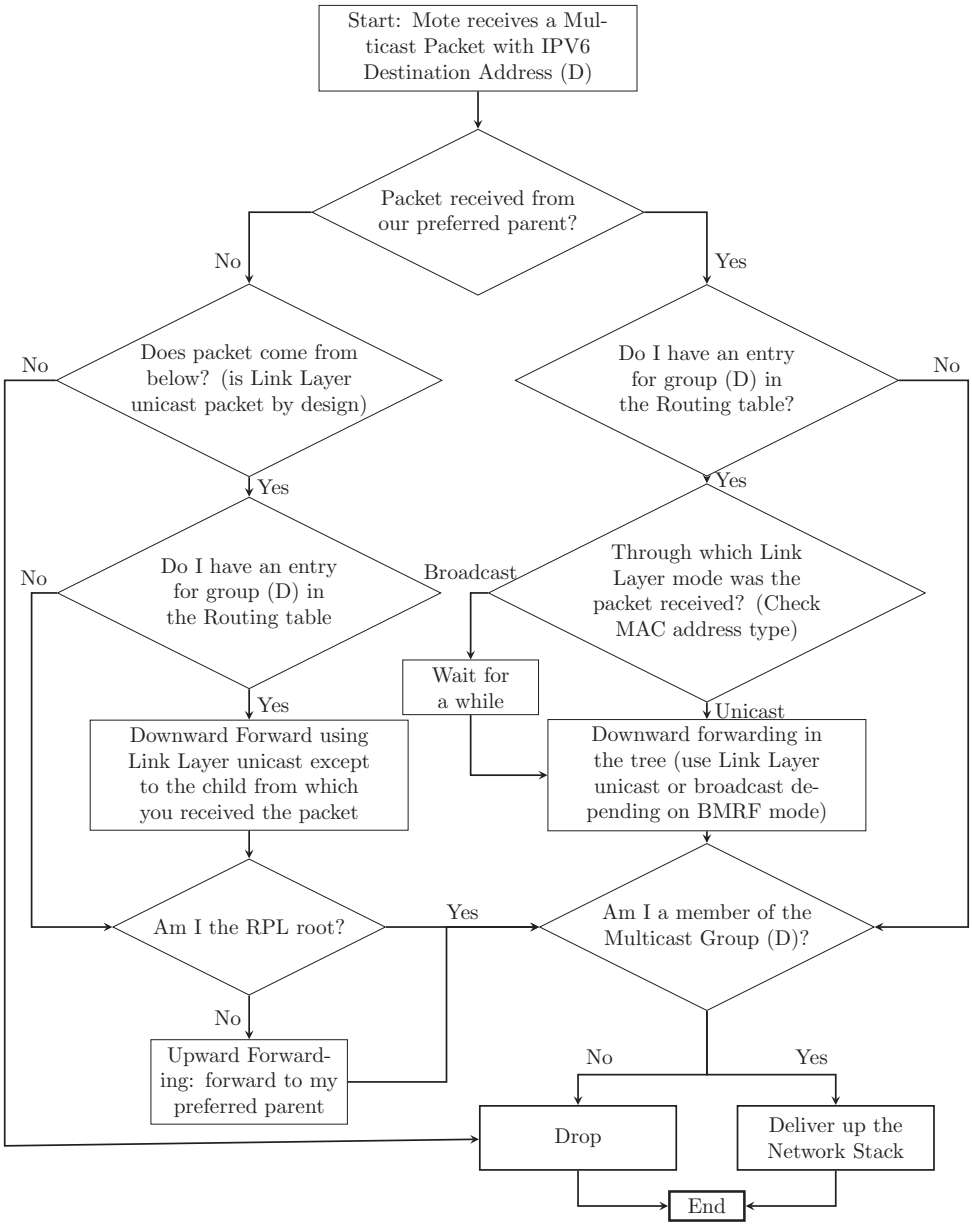

Fig. 9. BMRF forwarding algorithm.

is present for the respective multicast group, the packet is forwarded to all interested children, except to the one from which the packet originated. This forwarding is done through individual Link Layer unicast packets. If the mote is not the RPL root, the packet is also forwarded to the preferred parent using Link Layer unicast.

Why link layer unicast for downwards sending when the message come from below?. The same problems regarding duplicates emerge when forwarding the multicast packets. Packets sent upwards in the RPL tree must be processed only by the preferred parent of the sender. Since a router cannot distinguish whether the packet sender has selected it as its preferred parent, this filtering has to be done by the sender. Again, a Link Layer unicast offers the right solution.

The second problem is related to downward forwarding. When a packet comes from below in the tree (left side of the flowchart in Fig. 9), it should be forwarded to all interested children. A Link Layer broadcast transmission would be the best option when a lot of entries are found in the routing table. Unfortunately this would create duplicates. A router must avoid sending the multicast packet back to the child from which it has received that packet. If the mote would use a Link Layer broadcast, the child would again accept this multicast packet, although that packet had already been delivered in the respective sub-tree. Since the protocol is preferred to be stateless, there is no mechanism that allows motes to distinguish duplicate packets. This process is illustrated in Fig. 10.

As shown in Fig. 10, mote 19 forwards the multicast packet coming from source mote 22 up to its preferred parent mote 2. As mote 2 has three interested children, a Link Layer broadcast could be more efficient than three separate unicast transmissions to forward the packet to its children. However, mote 19 would also receive the packet again, and, as mote 2 is its preferred parent, it would accept the packet and forward it downwards, creating duplicates. The issue here is that mote 19 cannot know that the broadcast packet coming from mote 2 is the same packet that it has forwarded previously.

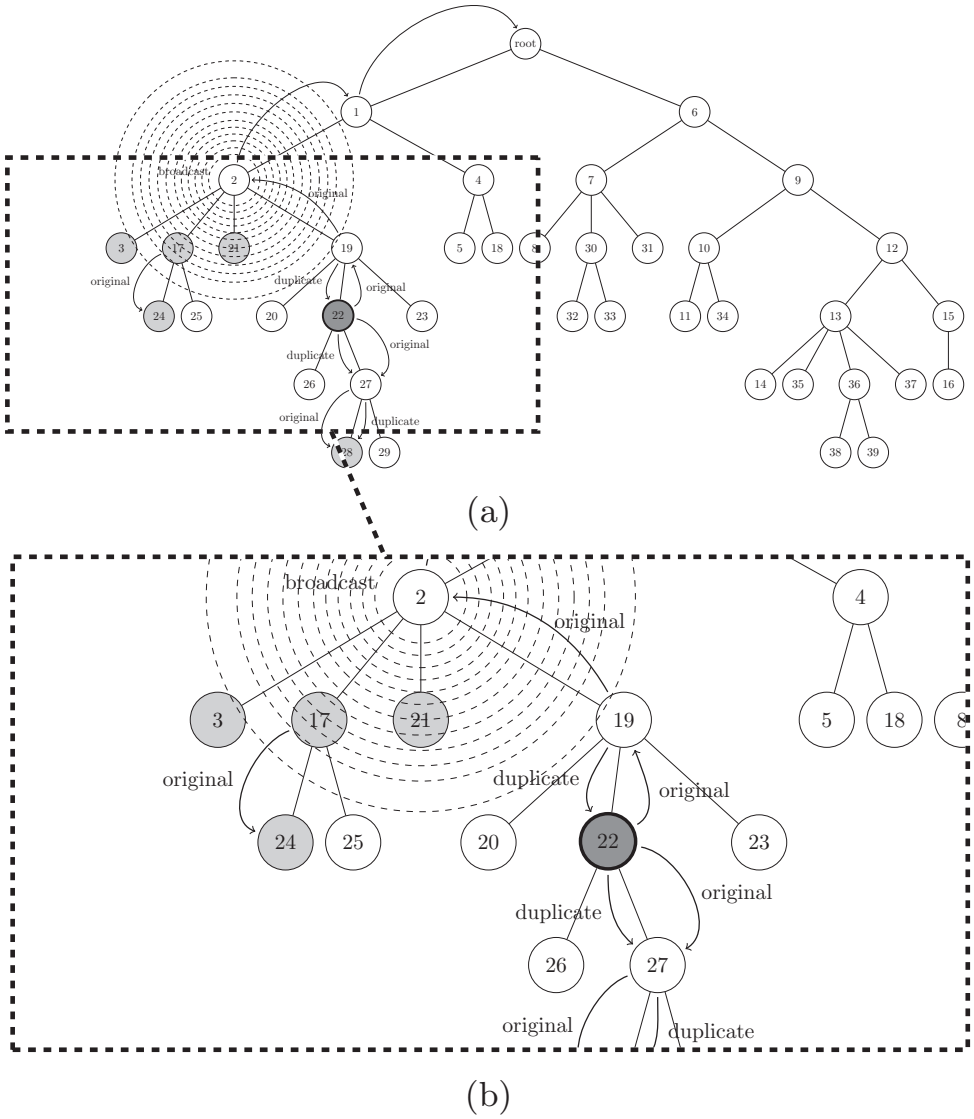

Fig. 10. Duplicates when forwarding with LL broadcast in BMRF.

3.2.3. Mode selection for downwards (sending or) forwarding

There are two ways to send a packet downwards in the tree. A Link Layer unicast can be used to send the packet to all interested children, which results in one packet being sent for each child. On the other hand, a Link Layer broadcast could be used to reach all neighbour nodes with one packet. However, because of interactions with RDC protocols, a broadcast will result in a series of packet transmissions, except in the case when RDC is switched off. In order to deal with this choice, three modes are available in BMRF:

Unicast mode. In Unicast mode, BMRF behaves similar to RPL Multicast. When a multicast packet is accepted for downward forwarding, the routing tables are checked. The router will send one Link Layer unicast frame to each of the interested children found in the tables.

Broadcast mode. In Broadcast mode, SMRF is imitated. When a multicast packet is accepted for downward forwarding, the routing tables are checked. If one or more entries are found for the present multicast group, a single Link Layer broadcast transmission is made whenever possible.

Mixed-T mode. In this mode, if the number of interested children is greater than a configurable threshold denoted *T*, the transmission is done with a Link Layer broadcast (as in Broadcast Mode), otherwise multiple unicast transmissions are used (as in Unicast Mode). This process is shown in Algorithm 1.

Algorithm 1: Mixed mode decision algorithm.

```
Event: IPv6 Multicast packet scheduled for forwarding;
begin
    if accepted then
        if interested_children >THRESHOLD then
            forward down with LL broadcast;
        else
            forward down with LL unicast;
        end
    else
        drop packet;
    end
end
```

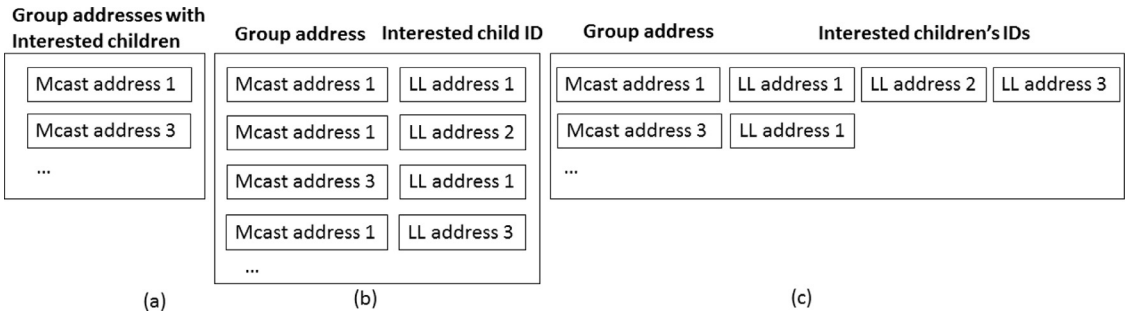

Fig. 11. Example of information kept for multicast routes for (a) SMRF, (b) BMRF simple implementation, (c) BMRF linked list implementation.

The optimal decision threshold regarding Link Layer unicast or broadcast mainly depends on the characteristics of the used RDC. For ContikiMac as RDC, an estimation of the optimal threshold for the minimum number of children needed for a broadcast to be more efficient can be calculated using Eq. (1).

$$T \simeq \frac{Max(\#radio\,transmissions\ for\ one\ Link\ Layer\ broadcast)}{Average(\#radio\,transmissions\ for\ one\ Link\ Layer\ unicast)} \quad (1)$$

The average number of radio transmissions for one Link Layer unicast is experimentally determined and is around 5. However, the number of radio transmissions for a Link Layer broadcast is proportional to the CCI and thus inversely proportional to the number of times a mote wakes up every second, called the channel check rate (CCR). In order to take this into account, we can use the values for a specific CCR, to calculate the values for other CCRs, using Eq. (2).

$$T \simeq \frac{Max(\#radio\,transmissions\ for\ one\ Link\ Layer\ broadcast)(for\ a\ specific\ CCR)}{Average(\#radio\,transmissions\ for\ one\ Link\ Layer\ unicast)} \times \frac{specific\,CCR}{choosen\,CCR} \quad (2)$$

Experiments show that the number of radio transmissions, with a payload of 40 bytes, in broadcast equates to 30 when the channel check rate equals 8 Hz. This results in Eq. (3) to calculate the optimal threshold.

$$T \simeq \frac{48}{CCR(Hz)} \quad (3)$$

3.3. Implementation choices and scalability issues

3.3.1. Extension in mote for multicast routing

SMRF only keeps track of the fact if there are any interested children, as shown in Fig. 11(a), but does not keep their identity as it always uses Link Layer broadcast transmissions anyway and does not support unsubscription. Since BMRF also suports Link Layer unicast transmissions as well as dynamic group management, this additional information should be stored in the mote.

In our implementation, we opted for adding the Link Layer address of the interested child to the route structure. This is done by adding a new routing entry for each child that is interested in the respective multicast address, as shown in Fig. 11(b). Since it is now possible to have multiple entries in the routing table for the same multicast group, we must avoid sending multiple DAOs for the same multicast group.

Improvements. By storing each interested child as a new routing entry, the multicast address is also stored multiple times. This can be prevented by storing multiple destination addresses in one single routing entry, as can be seen in Fig. 11(c).

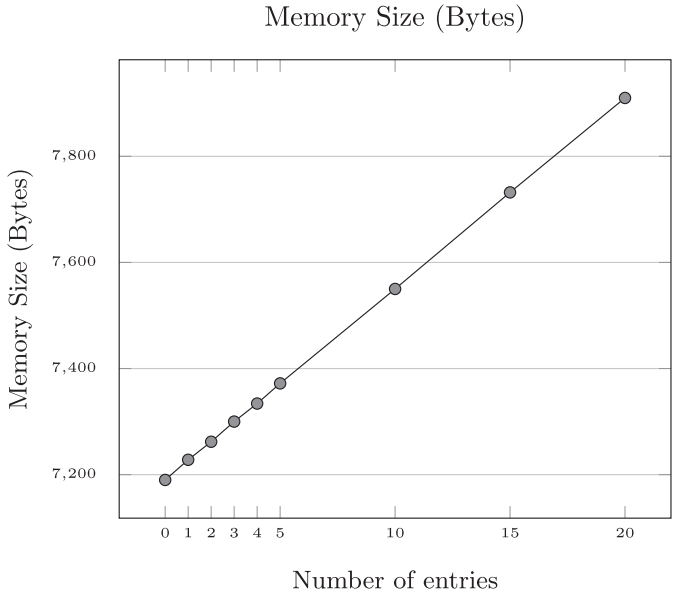

Fig. 12. Memory consumption.

This can at least save the size of an IPv6 multicast address (16 bytes) for every additional interested child, for a specific multicast address.

However, to do this efficiently, dynamic arrays are needed to avoid reserving unnecessary resources during the initialization of the routing table. It would be a big waste for the protocol to reserve multiple destination spaces for each entry. Since Contiki version 2.7 does not support dynamic arrays, this method could not be used. However, it will be implemented for the port to Contiki 3.0.

Memory consumption analysis. Since the protocol is now placing multiple entries in the routing table for a single IP multicast address, the maximum number of entries in the routing table should be expanded. The new size of the routing table should depend on the memory usage of the application, in order to maximize the number of possible routes.

Fig. 12 shows the influence of the maximum number of entries in the routing table on the RAM consumption. The relation between the maximum number of entries and the RAM consumption is linear. Adding an extra entry results in a RAM consumption increase of 36 bytes. This is higher than the 16 bytes of an IPv6 address, because a routing entry often stores additional information.

BMRF uses slightly higher ROM consumption, compared to SMRF. A program, with a simple application that sends multicast packets, uses between 42,445 bytes (broadcast mode) and 42,543 bytes (mixed mode), compared to 41,543 bytes for SMRF. This in-

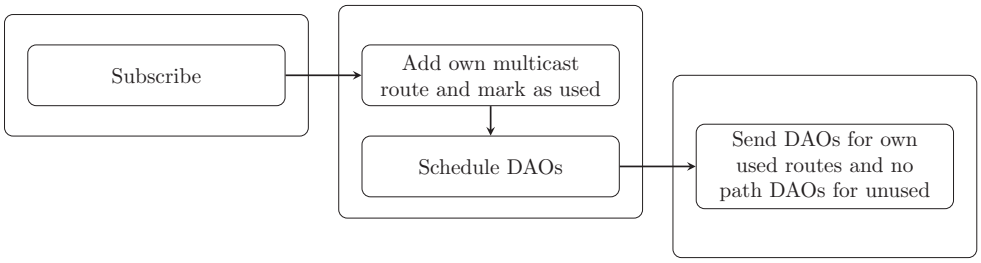

Fig. 13. Multicast group subscription process.

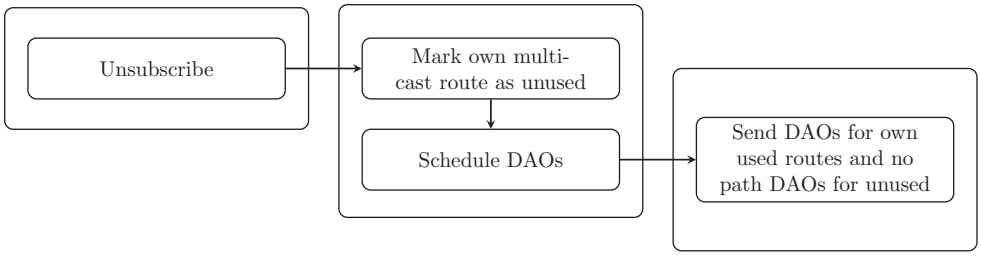

Fig. 14. Multicast group unsubscription process.

crease is an effect of the additional decision algorithms used in BMRF. However, it stays small enough to be used in the majority of sensor motes.

3.3.2. Unsubscription and route deletion

SMRF does not support unsubscribing from a multicast group, so changes were made to Contiki's RPL implementation to achieve this.

First of all, it is needed to prevent no-path DAO packets, i.e. with a lifetime value of 0, to be forwarded until no more routing entries for the multicast group are found. Otherwise, a parent might prune the multicast distribution tree while some of its children are still subscribed to that group.

Secondly, Contiki's RPL only sends DAOs for active multicast groups. This was changed to create no-path DAOs with a lifetime value of 0 for unused groups.

Lastly, the route purging mechanism was slightly modified. Whenever a stale route is detected, the route is removed. However, it might also be necessary to trigger a no-path DAO to reduce the multicast distribution tree.

The multicast subscription and unsubscription processes are illustrated in Figs. 13 and 14 respectively.

3.3.3. Scheduling DAOs

A final problem is, that changes in multicast subscriptions do not trigger a DAO renewal. This only happens when a Trickle timer expires or when an inconsistency is found. This means that changes in multicast group subscriptions are disseminated only after the Trickle timer expired, which could take a considerable amount of time.

Therefore, a new function has been created to trigger DAO renewal whenever a node subscribes to or unsubscribes from a multicast group.

4. Evaluation

In order to evaluate the performance of BMRF and compare it with SMRF, numerous simulation experiments were performed with Cooja.

Table 1
Simulation configuration (the sinks are the consumers of packets, namely the subscribers to the multicast group).

Motes	51 Sky motes (including root)
Subscriptions	25% (12 sinks), 50% (25 sinks), 75% (38 sinks) and 100% (50 sinks)
Radio medium	Unit disk graph medium (UDGM)
Ranges	transmit 50 m, interference 50 m
Topologies	Two: Random positioning in a 200 m × 200 m square
PHY and MAC	IEEE 802.15.4 with CSMA
RDC	ContikiMAC (CCR = 16 Hz, 32 Hz)
Iterations	4 for each parameter permutation
RNG seeds	New seed each iteration
Traffic	0.33 pkt/s - 100 packets
Payload	40 bytes
SMRF parameters	Contiki (Version 2.7) default configuration
BMRF parameters	Broadcast, Unicast and Mixed mode (Threshold: 1–4)

4.1. Scenarios

Different scenarios with varying number of motes subscribed to the multicast group were used to observe the behavior of the two protocols in different situations. Four settings were defined with respect to the size of the multicast group: 25%, 50%, 75% and 100% of the total number of motes subscribed to the multicast group.

Two different topologies created with a random positioning of the motes (removing topologies with orphaned motes) were considered to yield the most realistic results. Each setting was run multiple times, with a new random seed and randomly chosen motes subscribed to the multicast group for each iteration.

Fig. 15 shows an example of a RPL tree for one of these topologies. One should note that placing motes randomly results in a deep tree, for which most motes have a rather small number of children.

Since SMRF can only deliver multicast traffic originating from the RPL root, only scenarios in which the root of the tree is the source of the multicast traffic were taken into account for the performance evaluation.

With a channel check rate of 16 and 32, the threshold evaluates, using Eq. (3), to 3 and 2 respectively. Therefore only values around this optimized threshold are tested.

An overview of the configuration settings used in the simulations is given in Table 1.

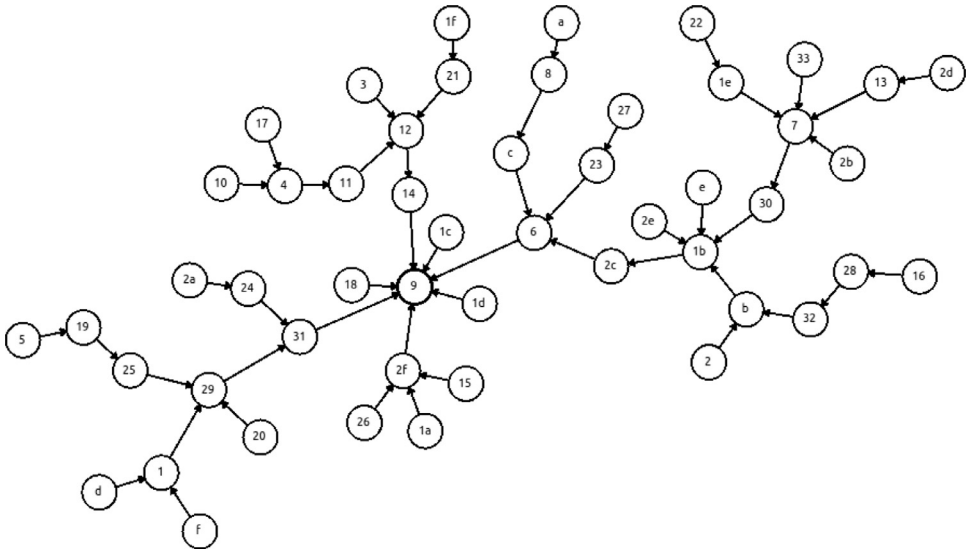

Fig. 15. Example of an RPL tree, formed in one of the random topologies, during simulation.

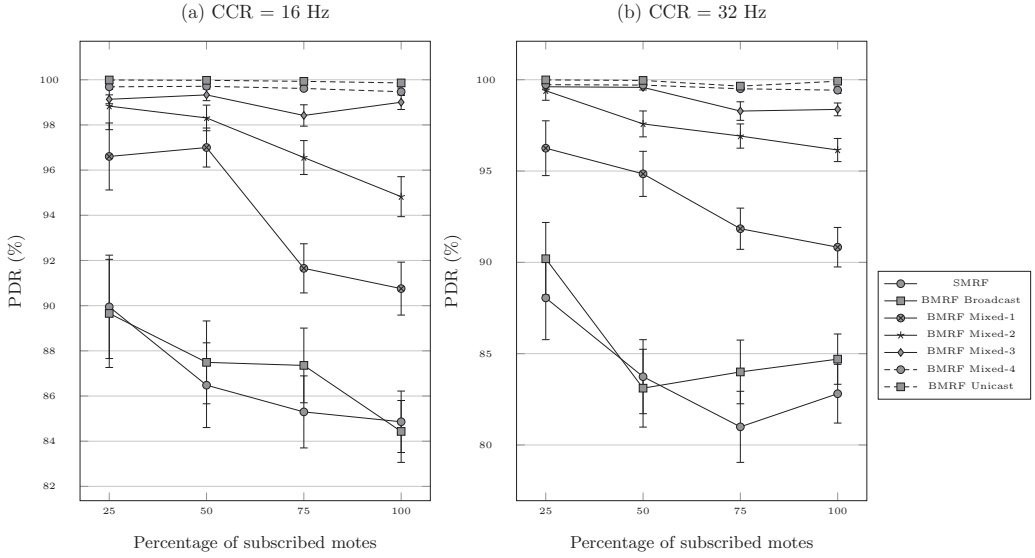

Fig. 16. The dotted lines show IPv6 multicast protocols, that are favoring unicast in the Link Layer, resulting in better values for packet delivery ratio. The best results for PDR are obtained with BMRF unicast, yielding a PDR of almost 100%, and the worst results are obtained with SMRF and BMRF broadcast, for which PDR has gone down to around 85%. A larger percentage of subscribed motes has a negative influence on the PDR. The channel check rate of ContikiMac has no significant influence on the PDR.

The following metrics were collected during each simulation after the stabilisation of the RPL tree (At that moment, the traffic was injected in the network and the measurements started):

- PDR (Packet delivery ratio)
- Number of packet transmissions (the number of requests from the CSMA layer to send a packet.)
- Number of radio transmissions (the actual number of packets sent by the radio driver. This number might be different from the number of packet transmissions, due to repeated sending of messages by the RDC layer.)
- Energy consumption per delivered packet (this is the total energy consumption divided by number of received packets.)
- End-to-end delay

4.2. Discussion of results

This section discusses how the percentage of subscribers influences PDR, number of packet transmissions, number of radio transmissions, energy consumption and end-to-end delay for SMRF and the different modes of BMRF, being BMRF broadcast, BMRF unicast, BMRF Mixed-1 till Mixed-4, for two different CCRs.

It should be noted that in the figures, shown in this section, dotted lines represent results from IPv6 multicast protocols that use, only unicast transmissions in the Link Layer or stick to unicast until more than four interested children. The solid lines represent the results obtained with IPv6 multicast protocols using, or only broadcast in the Link Layer, or sticking to broadcast until less than 4 interested children.

One should pay attention to the performance of the mode with optimized threshold (3 and 2 for CCR of 16 Hz and 32 Hz respectively), which should yield a very good performance for Mixed-3 and Mixed-2 respectively.

4.2.1. Packet delivery ratio and end-to-end delay

Fig. 16 shows that for each multicast group size, BMRF has a better PDR compared to SMRF. BMRF broadcast and SMRF have a comparable behavior. SMRF shows lower PDRs, as low as 85%, due to the dense topology and the collisions taking place using Link

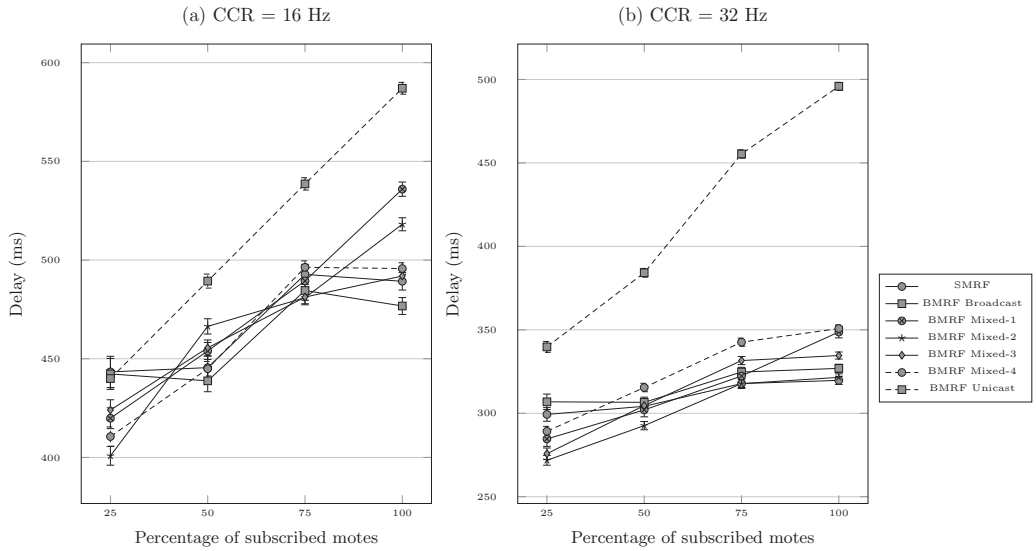

Fig. 17. The dotted line showing BMRF unicast, clearly shows the worst results for end-to-end delay. The best results for end-to-end delay are obtained by protocols mixing Link Layer broadcast and unicast. A larger percentage of subscribed motes also has a negative influence on the end-to-end delay, in particular for the protocols favoring Link Layer unicast. A higher channel check rate of ContikiMac results in a lower delay in general.

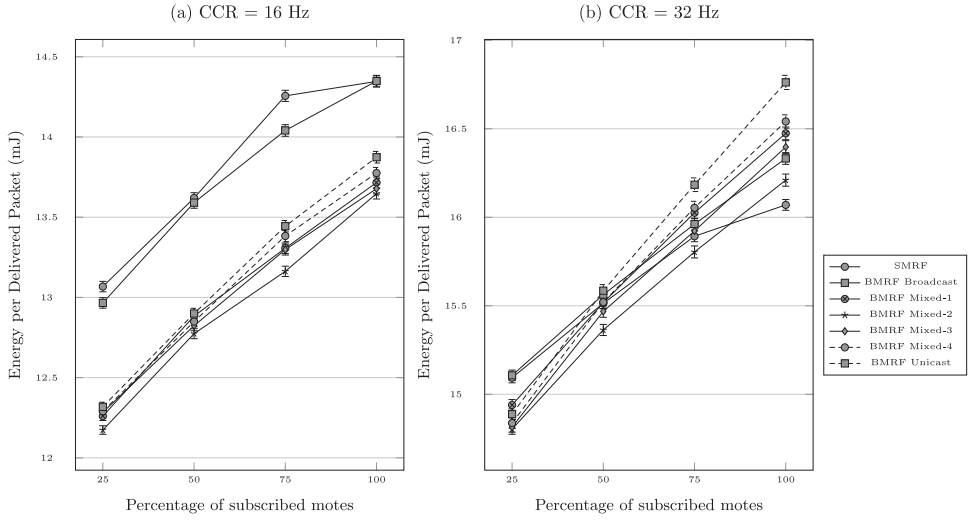

Fig. 18. The dotted lines that are showing IPv6 multicast protocols, that are favoring unicast in the Link Layer, clearly show worse results for energy consumption when 100% of the motes is subscribed to the multicast group. When more motes are subscribed, unicast gets less energy efficient. For multicast groups of 50 and 75%, the best results for energy consumption are obtained from BMRF Mixed mode, with a threshold of 2. Doubling the CCR results in a 15% increase in energy consumption.

Layer broadcast mode, for which packets are generated by the RDC layer, during the full duty cycle. It must also be remarked that unicast packets are acknowledged and the CSMA layer will retransmit a packet caught in a collision (up till three retransmissions are foreseen before dropping the packet), which greatly improves the chances for successful packet delivery.

For the end-to-end delay, it can be seen from Fig. 17 that the optimized thresholds perform better than SMRF and much better than BMRF unicast. Doubling the CCR, leads to a decrease of around 30% in delay for the best modes. Consequently, taking into account the PDR and delay properties, the BMRF threshold modes are able to offer better delays for a reasonable PDR (around 95% and higher).

4.2.2. Energy consumption

The energy consumption of all protocols is compared in Fig. 18. Fig. 18(a) shows the high energy consumption of SMRF and BMRF broadcast. For many subscribed children, the unicast mode also starts to consume more energy, in particular in Fig. 18(b), due to the shorter CCI. As can be seen, the mixed modes perform very good in all situations.

Fig. 18 also shows that increasing the number of subscribers in the network, results in a higher energy consumption, due to the increased probability to reach further away motes and the increased traffic. However, for SMRF, this effect flattens between 75% and 100% of subscribers, since for 75%, most of the tree already has to be reached. When the CCR doubles, the energy consumption increases with approximately 16% for the best modes.

Fig. 19 shows that if the CCR doubles, the amount of radio transmissions by SMRF halves, whereas the amount of radio transmissions by BMRF unicast stays the same. This is logical since broadcast forces packets to be sent by the RDC layer during the full duty cycle.

Packet transmissions and energy consumption are closely related, although, in Fig. 18(b), the unicast based modes consume more energy than the broadcast modes, even when the number of packets sent is lower. This can be explained by the fact that unicast

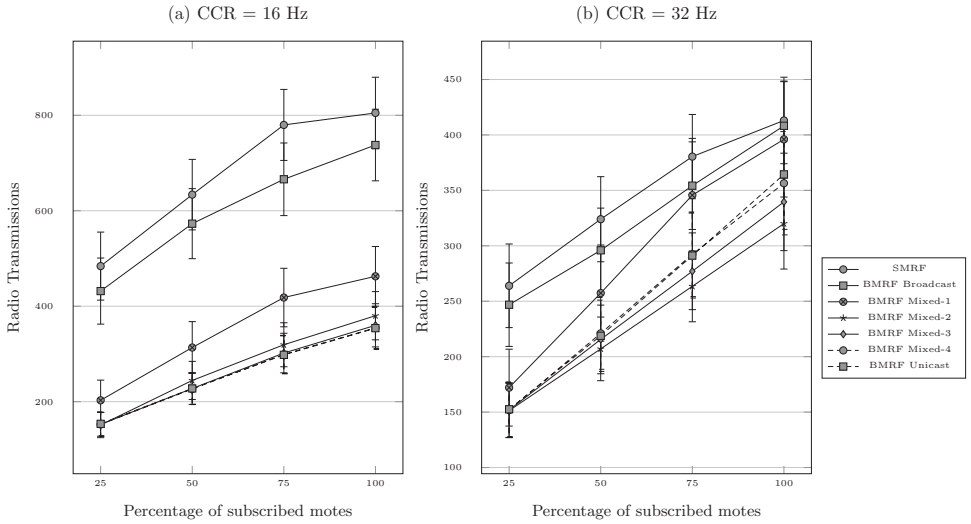

Fig. 19. The dotted lines show IPv6 multicast protocols, that are favoring unicast in the Link Layer, resulting in a lower number of radio transmissions, compared to protocols that only use Link Layer broadcast. The best result in terms of radio transmissions is obtained from BMRF Mixed mode with threshold 2 and the worst results are obtained from SMRF and BMRF broadcast. A larger percentage of subscribed motes has a negative influence on the number of radio transmissions. A higher channel check rate for ContikiMac has a positive influence on the number of radio transmission, especially for protocols using Link Layer broadcast.

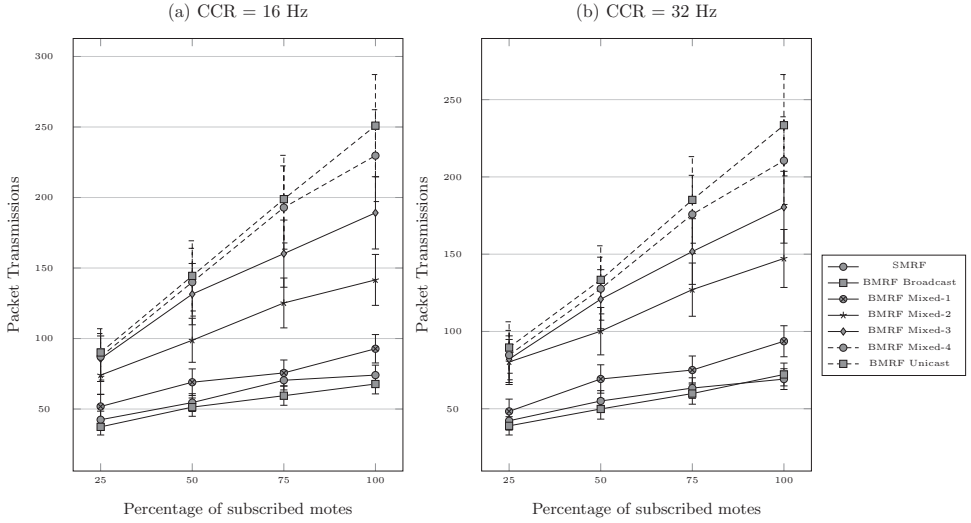

Fig. 20. The dotted lines show IPv6 multicast protocols, that are favoring unicast in the Link Layer, resulting in more packet transmissions. The worst result is obtained for BMRF unicast. The lowest number of packet transmissions are obtained, for all subscription percentages, by SMRF and BMRF broadcast and the better results are obtained from protocols favoring Link Layer broadcast. A larger percentage of subscribed motes also has a negative influence on the number of transmitted packets, in particular for the protocols favoring Link Layer unicast. The channel check rate of ContikiMac has no significant influence on the PDR.

keeps the radio active to wait for an acknowledgment after sending. The more packets, the more waiting for an acknowledgment.

In Fig. 20, one can see that channel check rate has almost no influence on the charts of packet transmissions at CSMA level. It is also clear that broadcast sends less packets at CSMA level. However, CSMA packet sendings can result many radio transmissions, because of interactions with the RDC layer (the smaller the channel check rate, the more radio transmissions).

5. Conclusion

We have presented BMRF, a multicast forwarding algorithm for IPv6 based WSNs, which addresses some of the shortcomings of the currently available solutions. In particular, our mechanism allows sources of multicast traffic to be located inside the network and support dynamic group registrations, at the expense of a slightly higher memory consumption.

Moreover, the proposed protocol is configurable in order to trade off energy consumption, latency, and reliability. Our experiments show that the proposed threshold for BMRF Mixed mode succeeds in getting the best of Link Layer broadcast and Link Layer unicast. For random topologies, the Mixed mode only yields gain for channel check rates higher than 8 Hz. For 8 Hz or less, the topology should be less randomized, but with parent motes with a lot of children (6 or more).

When reliability is crucial, BMRF unicast is the best choice, at the expense of a higher delay and energy consumption.

Acknowledgements

This research has been supported by the Agency for Innovation by Science and Technology in Flanders (IWT). The authors are also thankful to the COST Action IC1303 AAPELE.

References

[1] J.W. Hui, D.E. Culler, IP is dead, long live IP for wireless sensor networks, in: Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems, in: SenSys '08, ACM, New York, NY, USA, 2008, pp. 15–28.

[2] J. Zheng, A. Jamalipour, Wireless sensor networks: a networking perspective, in: The Oxford Handbook of Innovation, John Wiley & Sons, 2009, pp. 145–172.

[3] N. Kushalnagar, G. Montenegro, C. Schumacher, IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals, RFC 4919, RFC Editor, 2007. http://www.rfc-editor.org/rfc/rfc4919.txt.

[4] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, R. Alexander, RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, RFC 6550, RFC Editor, 2012. http://www.rfc-editor.org/rfc/rfc6550.txt.

[5] P. Porambage, A. Braeken, C. Schmitt, A. Gurtov, M. Ylianttila, B. Stiller, Group key establishment for enabling secure multicast communication in wireless sensor networks deployed for iot applications, IEEE Access 3 (2015) 1503–1511.

[6] A. Rahman, E. Dijk, Group Communication for the Constrained Application Protocol (CoAP), RFC 7390, RFC Editor, 2014.

[7] I. Ishaq, J. Hoebeke, I. Moerman, P. Demeester, Experimental evaluation of unicast and multicast coap group communication, Sensors 16 (7) (2016) 28, doi:10.3390/s16071137.

[8] G. Maia, A.L. Aquino, D.L. Guidoni, A.A. Loureiro, A multicast reprogramming protocol for wireless sensor networks based on small world concepts, J.f Parallel Distrib. Comput. 73 (9) (2013) 1277–1291, doi:10.1016/j.jpdc.2013.05.006.

[9] M. Antonini, S. Cirani, G. Ferrari, P. Medagliani, M. Picone, L. Veltri, Lightweight multicast forwarding for service discovery in low-power IoT networks, in: Software, Telecommunications and Computer Networks (SoftCOM), 2014 22nd International Conference on, 2014, pp. 133–138.

[10] X. Wang, Multicast for 6LoWPAN wireless sensor networks, IEEE Sensors J. 15 (5) (2015) 3076–3083, doi:10.1109/JSEN.2014.2387837.

[11] J. Hui, R. Kelsey, Multicast Protocol for Low power and Lossy Networks (MPL), RFC 7731, Internet Engineering Task Force (IETF), 2016.

[12] P. Levis, T. Clausen, J. Hui, O. Gnawali, J. Ko, The Trickle Algorithm, RFC 6206, RFC Editor, 2011. http://www.rfc-editor.org/rfc/rfc6206.txt.

[13] J. Hui, J. Vasseur, D. Culler, V. Manral, An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL), RFC 6554, RFC Editor, 2012.

[14] G. Oikonomou, I. Phillips, Stateless multicast forwarding with RPL in 6LowPAN sensor networks, in: Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on, 2012, pp. 272–277.

[15] G. Oikonomou, I. Phillips, T. Tryfonas, IPv6 multicast forwarding in RPL-based wireless sensor networks, Wireless Personal Commun. 73 (3) (2013) 1089–1116.

[16] A. Dunkels, The ContikiMAC Radio Duty Cycling Protocol, Technical Report, SICS, 2011.

[17] K.Q. Abdel Fadeel, K. El Sayed, ESMRF: enhanced stateless multicast RPL forwarding for IPv6-based low-power and lossy networks, in: Proceedings of the 2015 Workshop on IoT Challenges in Mobile and Industrial Systems, in: IoT-Sys '15, ACM, New York, NY, USA, 2015, pp. 19–24.

[18] J. Hui, J. Vasseur, The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams, RFC 6553, RFC Editor, 2012. http://www.rfc-editor.org/rfc/rfc6553.txt.

Guillermo Gastón Lorente is currently a master student at the Illinois Institute of Technology (Chicago, IL), for obtaining the degree of Master of Science in Computer Engineering, jointly with a Master degree in Telecommunications Engineering from the Universidad Politécnica de Madrid.
In 2014 he obtained the bachelor degree from Universidad Pblica De Navarra (UPNA) for which he realised his bachelor thesis at Vrije Universiteit Brussel (VUB) in the ETRO-Smartnets research group in the context of an Erasmus exchange between UPNA and VUB. From Universidad Pública De Navarra he received an outstanding graduation award in 2015.
Today he still collaborates with the smartnets group on the topic of multicast in Wireless Sensor Networks, in particular with PhD student Bart Lemmens. His current interests are network, protocol and software design, cross-layer protocol optimization, Wireless Sensor Networks and cyber-security.

Bart Lemmens has received his M.Sc. in Industrial Sciences with specialization electronics-ICT from the Erasmushogeschool Brussel, Belgium in 2008. He is currently pursuing his Ph.D. degree in Engineering Sciences at the Department of Electronics and Informatics from the Vrije Universiteit Brussel. His research interests include energy efficient and multi-channel Medium Access Control protocols for Wireless Sensor networks, multi-agent systems, and time synchronization in networks.

Matthias Carlier received a master in Industrial Engineering in 2013 from Vrije Universiteit Brussel (VUB). Currently he is a PhD student at the department of Electronics and Informatics (ETRO) and the department of Industrial Engineering (INDI) at Vrije Universiteit Brussel (VUB). His research interests include the design, implementation and evaluation of routing protocols for Wireless Sensor Networks for building automation, environmental monitoring, autonomous ground vehicle applications and smart grids.

An Braeken received the M.Sc. degree in mathematics from the University of Gent, in 2002, and the Ph.D. degree in engineering sciences from the Computer Security and Industrial Cryptography Research Group, KU Leuven, in 2006. She worked for almost two years with BCG, a management consulting company. In 2007, she became a Professor with the Industrial Sciences Department, Erasmushogeschool Brussel, where she has been with Vrije Universiteit Brussel since 2013. Her current interests include cryptography, security protocols for sensor networks, secure and private localization techniques, and Field-programmable gate array implementations.

Prof. dr. ir. Kris Steenhaut received the master in Engineering Sciences in 1984 and the master in Applied Computer Sciences in 1986 and the PhD degree in Engineering Sciences from Vrije Universiteit Brussel (VUB) in 1995. Currently she is professor at the department of Electronics and Informatics (ETRO) and the department of Industrial Engineering (INDI), Faculty of Engineering Sciences, Vrije Universiteit Brussel, Belgium. Her research interests include the design, implementation and evaluation of Wireless Sensor Networks for building automation, environmental monitoring, autonomous ground vehicle applications and smart grids. She has cooperated in and coordinated European ITEA consortia such as ESNA (European Sensor Network Architecture) and ISN (Interoperable Sensor Networks) and has participated in FP7 under the ICT Policy Support Programme on the theme ICT for a low carbon economy and smart mobility. She is also a driving force in several strategic national research projects on the theme of machine learning for energy efficiency operation of devices and machines, on sound/air pollution monitoring and on water management by means of heterogeneous sensor networks. Her research output includes 150+ co-authored articles in international journals and proceedings of international conferences, as well as the organization of tutorials and workshops at international conferences.