

RESEARCH

Open Access

Support of multiple sinks via a virtual root for the RPL routing protocol

David Carels*, Niels Derdaele, Eli De Poorter, Wim Vandenberghe, Ingrid Moerman and Piet Demeester

Abstract

Data acquisition in large wireless sensor networks consisting of only a single sink can typically lead to scalability and energy efficiency issues. A solution to this problem is the deployment of multiple sinks in the network. This approach is however not supported by the popular sensor network routing protocol, IPv6 routing protocol for low-power and lossy networks (RPL). This paper describes a method to support the usage of multiple sinks for RPL in accordance to the limited guidelines of *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks* (RFC 6550). Hereby this paper shows that the concept of a virtual root can work and can be implemented with a minimal complexity. The correct behaviour of this extension was verified, by performance tests, in both a simulation environment and a real-life environment (iMinds wiLab.t office testbed). The chosen approach has the advantage that for an existing deployment of a RPL network, only the sink nodes need to be adapted. The results confirm that the use of multiple sinks in RPL can deliver the desired advantages. For an increase in the number of sinks from 1 to 4, a decrease of about 45% in the maximal and more than 30% in the average energy consumption was obtained in simulations for the used topology. For the real-life tests, the average energy consumption decreased with more than 30% and with more than 50% for the maximal energy consumption when the number of sinks was increased from 1 to 2 on the iMinds wiLab.t office testbed. By using a positioning algorithm to determine the optimal position, for the sinks, possibly even better performances can be obtained.

Keywords: RPL; Multi-sink; WSN; Routing; Virtual DODAG root

1 Introduction

In wireless sensor networks (WSNs), there will typically be one sensor node, referred to as the sink, responsible for collecting the data from a number of data sources. As the size of the network grows, the average number of hops between the data sources and the sink will become larger. This introduces more packet loss, higher energy consumption and thus a decrease in the lifetime of the sensor nodes. Therefore, it is needed to keep the distance or the number of hops between a sensor and its destination sink as small as possible. A solution to this problem is to deploy multiple sinks in a wireless sensor network. To increase the performance of the wireless sensor network (in terms of packet loss, lifetime,...), each sensor node communicates only with the nearest sink. If the sinks are

spread over different locations, this can result in a reduction of the average number of hops and thus an increase in performance and reduction of the consumed energy. Another advantage of a multi-sink approach is the reduction of the sink hole problem. Because multiple sinks are deployed, the traffic load can be spread over the different sinks which results in a reduction in the load of the nodes in the direct environment of the sink. This leads to an increase in the lifetime (time until the first node is out of battery source) of the network. Multiple sink support is also usable to connect infrastructure over different locations/interfaces in one network.

The IPv6 routing protocol for low-power and lossy networks (RPL), is designed by the Internet Engineering Task Force (IETF) ROLL (routing over low power and lossy networks) working group to overcome routing issues in low-power and lossy networks (LLNs). The goal was to design an adaptable standardised protocol for different application domains instead of a large amount of different (application specific) routing protocols. The protocol

*Correspondence: david.carels@intec.ugent.be
Department of Information Technology (INTEC), Ghent University - iMinds,
Gaston Crommenlaan 8 Bus 201, Ghent 9050, Belgium

is mainly intended for multipoint-to-point traffic and also supports point-to-multipoint and point-to-point traffic.

This paper contributes in the support of multiple sinks for the IETF RPL protocol by

- Showing how a virtual root can work in accordance to the RPL standard.
- Implementing support with minimal complexity.
- Demonstrating an easy to roll out support on existing infrastructure (only sinks need to be adapted).
- Quantifying the gain of support with a virtual root by simulations and real-life tests.

After evaluation by simulation and real-life tests, this paper demonstrates that multi-sink support for RPL, with a virtual root and a limited complexity, in accordance to the guidelines of the IETF RPL standard (RFC 6550) functioning.

In Section 2 an overview of relevant literature is given. The RPL routing protocol to which multi-sink support is added is discussed in Section 3. In Section 4 the problems which need to be solved are stated. Section 5 describes how the support for multiple sinks is implemented. The evaluation by simulation of the added support is given in Section 6. Thereafter, the results of the simulations are verified with some experimental tests on the iMinds wiLab.t office testbed described in Section 7. Section 8 tries to evaluate the effect of using multiple sinks on the point-to-point paths. Finally, in Section 9 the conclusions of this paper are made.

2 Related work

In multi-hop sensor networks, a significant part of the consumed energy is used for the forwarding of messages/packets. This energy consumption can be reduced by different techniques like minimising path length or hop count.

This minimisation also contributes to the increase of the network lifetime since the nodes in the neighbourhood of the sink are responsible to forward all the traffic from their subtree. Especially, the nodes one hop away from the sink have to forward the most of the messages. In [1] it was shown that in some cases, nodes in the network still

have 90% of their capacity while nodes in the direct neighbourhood of the only sink are exhausted. The greater the subtree of a node is, the more packets it has to forward to the sink.

A first technique to reduce the number of hops is the introduction of different sinks (see Figure 1).

2.1 Multi-sink support

The positive effect of the introduction has already been demonstrated in different research papers. In [2] the average number of hops a packet has to travel compared with the number of sinks has been researched. The paper demonstrated that the average number of hops a data packet has to travel decreases as the number of sinks increases. Together with a reduction of the average number of hops, the average energy cost also has decreased.

In [3] it is demonstrated that an increase in the number of sinks also results in an increase of the minimum data volume each sensor node can produce.

The influence that an increase in the number of sinks can have on the network lifetime of a sensor network is illustrated in [4]. The paper illustrates these effects by the comparison of the number of sinks to the exhausted and unreachable nodes over time. Both the percentage of exhausted and unreliable nodes have a smaller increase when the number of sinks is larger. The effect an exhausted node close to the sink will cause is that a larger amount of nodes get unreachable. This can also be deduced from the results of this paper.

Also in [5] the increasing lifetime for an increasing number of sink nodes is demonstrated. Linked with the increased lifetime of the network, the number of successfully received sensor readings at the sinks increases when the number of sinks is increased.

In [6], initially, an inversely proportional effect between the power usage and the number of base stations is demonstrated. However, when the network becomes saturated with more sink nodes, the added sink nodes yield smaller power savings.

All these papers demonstrate the positive effects on the energy consumption, on the lifetime of the network and/or on the throughput of information.

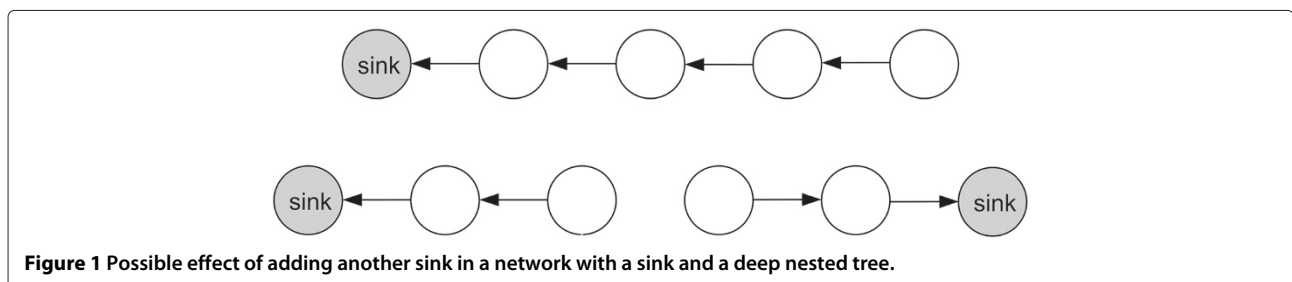


Figure 1 Possible effect of adding another sink in a network with a sink and a deep nested tree.

This paper however focuses on the advantages and techniques of multi-sink support for the IETF RPL protocol. To the best of our knowledge, there exist no implementation or performance tests of multiple sink support for RPL. Therefore, this paper wants to investigate the possibilities for the introduction in the RPL protocol.

2.2 Multiple sink placement

It is obvious that an optimal placement of the sinks is crucial to obtain an optimal efficiency gain. The determination of the optimal position of the sinks is already studied in several publications.

In [3] the formulation, via a linear programming model, to find the optimal position of multiple sink nodes and to find the optimal traffic flow from the different sensors to the sinks is proposed. This formulation has the objective to maximise network lifetime and to ensure fairness. The proposed scheme is compared to the multi-sink aware minimum depth tree (m-MDT) scheme, which shows that the proposed scheme outperforms the m-MDT scheme.

In [4] on the best sink location problem, the effect of exhausted nodes on the reachability of all nodes in the network is illustrated for different snapshots in time. Hereby is observed that the number of disconnected regions increases over time and the failure of nodes close to the sink, which are serving a large branch, also increases. So they conclude that the reconstruction of the minimum energy tree after the occurrence of energy failures can prolong network lifetime. The paper also proposed a solution technique for the MSPOP (minimise the number of sinks for a predefined minimum operation period) problem and simulated it for a random network. The simulations show a decrease in the percentage of exhausted nodes for an increase in the number of sink nodes. This reduction is linked to the reduction in cluster size and the shorter paths from each node to the sink for an increasing number of sink nodes. For a network lifetime with a ratio of 25% unreachable nodes after 1 or 2 months, respectively, four or six sink nodes are needed.

The positive effect of the deployment of multiple sinks and correct sink placement on the network lifetime are also illustrated in [5]. In comparison between topology aware and geo-aware placement strategies, the topology aware placement gives remarkable lifetime improvements to the geo-aware placement. The use of routing metric placement (RMP) results in lifetime increases of 60% for two sinks and 25% for three sinks compared to the use of KSP (*K*-means placement based on the *K*-means clustering algorithm [7]). SPP (shortest path placement) and RMP placement for two sink nodes even result in a better lifetime than the KSP placement of three sink nodes.

The problem of optimising the positioning of sink nodes or base stations to minimise the energy consumption of the sensors in the network, is also considered in [6]. By

simulation tests, the energy gain is demonstrated when using a local search algorithm to find the optimal positions for the sinks. In this local search algorithm, there was the restriction to position the sinks at sensor locations. The results were compared with a greedy algorithm and some random samples. The simulations were tested for a regular grid, a uniform random graph, and a random graph with preferential attachment. In all the scenarios, the local search algorithm delivered the most optimal power usage rate.

In [8] a mathematical model is used for the determination of the optimal position of the sink nodes. This model tries to find the optimal positions by minimising the sensors' average distance to the nearest sink. Based on the model, two positioning algorithms were proposed. In the global algorithm, the information of all the sensor nodes in the network is used and in the *1hop* algorithm only information of the neighbouring sensors is used. Despite that the *1hop* algorithm only uses local information, the average communication distances, compared to the global algorithm, are only a few percent higher. The *1hop* algorithm, however, has no scalability issues thanks to the usage of only local information. To balance the traffic load and to prevent depletion of nodes around the sink, the paper also presents the *1hop* relocation algorithm. Through simulations, a doubling of the network lifetime compared to the static deployment or random relocation was demonstrated. Compared to relocation based on global information, the *1hop* relocation achieves 30% higher lifetime.

In [1] the MSEP (multiple static sinks edge placement) model is discussed. In this model the multiple sinks are located on the same edge. In the MSEP model, different strategies for sending data to the sink are possible, namely, sending data to all the sinks (all sink notification), sending data to closest sink (closest sink notification) or sending data to interested sinks only (interested sink notification). Because in the first and last strategy the total amount of packets increases, the paper only focuses on the closest sink notification. In a first case of non-overlapping critical zones, they state that in a *n*-sink environment where the sinks serve an equal part of the network, the nodes in the critical zone deplete at the same time. However, for large networks still a large amount of energy is unutilised. If the sinks do not serve an equal part of the network, the network first will behave as one-sink networks which serve their part of the network and with equal initial energy capacity. Once the nodes in the critical zone of sinks get depleted, the network will behave as a network with only one sink with a smaller initial energy capacity. The adding of extra sinks on the same edge can lead to the overlapping of the critical regions. This situation is called the full-edge sink placement. In this situation still 50% of the energy is not used. These results also apply on

the multiple static sinks with random placement (MSRP) model for large networks with a relatively small number of sinks.

The multi-hop genetic algorithm (MH-GA) presented in [9] tries to balance the load for multiple sinks in a wireless sensor network. The obtained results are compared with the least delay (LD) binding approach. First the paper demonstrates that MH-GA outperforms LD in terms of less variance of remaining energy among sinks in a uniformly distributed network. This improvement can be explained because LD, in contrast to MH-GA, only takes the delay into account and does not perform any load balancing. For non-uniformly distributed networks, MH-GA still outperforms LD, but for both algorithms, the variance increases faster due to the difference in density in the network and possible areas where sensors only can bind to one sink which makes load balancing more difficult. The results of the paper show an improvement in the death time of the first sink, but also demonstrate that a good spreading of the load for the sinks can improve the lifetime of the first death sink with approximately 57% for uniform networks compared to the LD approach. For non-uniform networks the improvement is smaller, but MH-GA still results in a larger lifetime of the first failing sink. The smaller improvement originates also in the fluctuation of the density. These results also show that an optimal placing of the sink nodes contributes to the prolongation of the lifetime of the network. On the other hand, it should be noted that MH-GA, due to the load balancing, not always chooses for the solutions with the least delay.

This paper however does not search for the optimal positioning of sink nodes because it currently could not be verified for RPL, since the support for multiple sinks is currently lacking.

2.3 Mobile sinks

Another technique to prolong the lifetime of the network is the use of mobile sink nodes to balance the traffic load for the different nodes of the network. In [10-14] it is demonstrated that mobile sinks can increase network lifetime by preventing saturation on nodes which are close to the fixed sink. However, the use of mobile sinks is out of the scope of this paper.

3 Routing protocol for low-power and lossy networks

3.1 Overview

RPL [15] is a proactive, distance-vector routing protocol specifically designed for LLNs and optimised for multi-point to point traffic. Some characteristics of LLNs are high bit error rates, low bandwidth and instability. RPL has been developed with four different scenarios in mind: i.e. urban networks, building automation, industrial

automation and home automation. RPL forms a tree-like topology rooted at the sink, called a destination-oriented directed acyclic graph (DODAG). Directed acyclic graphs (DAG) have the property that all edges are oriented in such a way that no cycles exist. If all edges are contained in paths oriented towards a single node, the DAG is called *destination-oriented*. Since different applications have different needs, RPL only specifies how to build a DODAG, and the characteristics of the DODAG (e.g. criteria to choose a parent) are specified by an objective function. Additionally, RPL operations require bidirectional links, which can be asymmetric.

Although link-state routing protocols are more powerful if the whole network topology is known, for RPL a distance-vector routing protocol is used. The choice for a distance-vector routing protocol is based on limited capabilities (limited memory and processing power) of sensor nodes.

3.2 Objective function

In RPL the routing structure is constructed based on the objective function. This objective function specifies how to compute the rank of a node. The rank is computed, based on the hop count from the DODAG root to the node and on metrics and/or constraints. This rank depicts the relative distance to the DODAG root. These objective functions can be adapted to the specific needs of the network and/or applications and can vary from simple (for example only hop count) to complex functions.

A link metric that is often used is the expected transmission count (ETX). The ETX value of a link is the expected number of transmissions that is required to successfully send a packet over that link.

3.3 Topology formation

The DODAG is built based on the information of DODAG information object (DIO) messages (ICMPv6 messages defined in RPL). These messages contain a DODAGID, rank information and objective function. DIO packets are first sent by the root and then periodically by each node of the DODAG after calculation of its own rank. The rate at which the DIO messages are being sent is dynamically tuned, using the Trickle algorithm [16]. In the absence of changes in the DODAG structure, the algorithm doubles the period of the DIO messages after each transmission of a DIO message; otherwise, the trickle timer is reset to send the DIO messages more frequently in order to propagate the updated DODAG quickly.

When a node receives a DIO from a neighbouring node, the sender is inserted in a list of possible successors. Based on the rank of a RPL node (relative position in the DODAG) and its metric value, a decision is made by the objective function whether or not to use a node as preferred parent. All the traffic the node has to send to the

DODAG root or messages which are not directly routable by the node are sent to this preferred parent.

To enable downwards traffic, an RPL node sends a destination advertisement object (DAO) message (also an ICMPv6 message) to its preferred parent to advertise prefix reachability towards the leaves.

Since the period between two DIO messages can become very large in stable networks and as a mechanism to explore a new network environment, it is also possible for a node to solicit for DIO messages by sending link-local multi-cast DODAG information solicitation (DIS) messages.

4 Problem statement

4.1 Multiple DODAGs versus virtual sink

A first thought to support multiple sinks can be to construct a DODAG for each sink. Support for multiple DODAGs with different roots is offered in the specifications of RPL. However, the use of multiple DODAGs would result in trees in which all the nodes of the network are present for each different root. However, when using a single DODAG with a virtual root which coordinates multiple LLN sinks, only one DODAG is constructed and the different sinks will only have a subset of the nodes in their subtree. These subtrees are interconnected via a virtual sink which can be seen as the parent of all the sinks. All these subtrees together form one tree including all the nodes of the network.

Firstly, the first solution differs in memory required. When using multiple DODAGs, each sink has to store the routes to each node in the network, and the subtrees of the child of each sink will be bigger, which also results in higher memory usage for the children of the sinks. This memory usage per sink is similar to the memory usage when using only one sink. When using a virtual sink, the memory usage to store the routes to children in the subtrees will be spread over the different possible sinks. However, the placement of the sinks will influence the number of children per sink.

Secondly, the two solutions will differ in terms of processing needed. When using different DODAGs, the nodes in the network need extra processing for each extra DODAG instead of processing for only one sink, when using a virtual sink. Another point of difference is the extra communication needed for each additional DODAG. To maintain each additional DODAG, each node needs to send DIO and DAO messages to maintain the tree of each DODAG. However, due to the Trickle algorithm, the number of messages needed will decrease exponentially for stable networks.

So the first solution requires extra memory, extra processing and extra communication for each sink added. This is an even worse solution than the initial situation with one sink because now not only the nodes close to the

original sink are heavy loaded, but also the nodes close to the additional sinks, and there are four times more control messages. We think the usage of multiple DODAGs is less suitable to solve the problem. Multiple DODAGs are more suitable for situations where different objective functions are used to optimise the network based on different requirements (minimise packet loss, latency, hop count,...).

The last solution requires less memory usage, less processing and less communication in the different sinks. This solution constructs a DODAG where the level 0 exists of a virtual node and the level 1 exists of all the sink nodes. If the sinks are positioned as good as possible, the load and nodes are divided between the different sink nodes. However, some extra communication is needed to synchronise the different sink nodes.

4.2 RFC specification for multiple sinks

Current support for multiple sinks in the RPL specification [15] is limited to the specification of 'a single DODAG with a virtual root that coordinates LLN sinks (with the same DODAGID) over a backbone network'. The specification gives the example of 'multiple border routers operating with a reliable transit link, e.g., in support of an IPv6 Low-Power Wireless Personal Area Network (6LoWPAN) application, that are capable of acting as logically equivalent interfaces to the sink of the same DODAG'. However, the method of coordination is out of the scope of the request for comment (RFC). This paper used the vision of the RFC to implement and to evaluate the use of multiple sinks in RPL.

As stated in the RPL specifications [15], the usage of multiple sinks will require reliable communication between the sinks. This requirement however is not that restrictive, because these sinks normally are already connected to a reliable network to report their data to a database or central application. The biggest challenge to enable multi-sink support for usage with RPL is the coordination and synchronisation of the different sink nodes. This coordination and synchronisation is needed to keep the different sink nodes in a common state to act together as a virtual sink.

This synchronisation and coordination will also influence the routing of data, especially point-to-multipoint and point-to-point traffic which are related to each other.

4.3 The RFC concept of multi-sink

To support multiple sinks, the concept of a virtual sink node is proposed (see Figure 2) for the implementation. This means that the actual sinks will behave as if they were a child of a common single sink. Since communication between the sinks is crucial, a wired network is used instead of the wireless medium. If a real physical sink receives a packet destined for the virtual sink, the packet

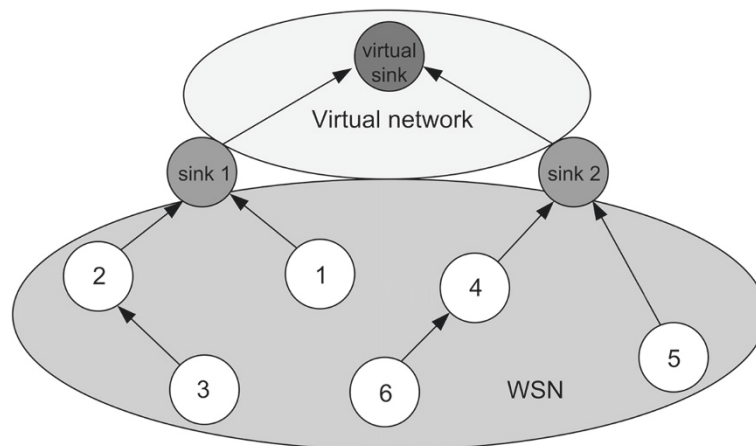


Figure 2 Example of a DODAG with two real sinks and one virtual sink.

will not be forwarded because the virtual sink does not exist, but the real sink will handle the packet himself and take appropriate actions.

Due to the use of a virtual sink, the non-sink nodes inside the sensor network do not know that there are actually multiple sinks inside the network. The additional advantage is that there are no changes or reconfigurations needed to the existing nodes in the network.

4.4 Coordination between sink nodes

The support of multiple sinks, as described, implies that there will no longer be only one physical node on top of the DODAG. Each physical sink will be responsible for the communication to and from the nodes in its subtree. This has the disadvantage that a sink will not be able to communicate directly with each node inside the network. The advantage is that the routing tables for each sink will be reduced. The reduction of the number of routing entries can help to reduce the number of nodes which are not routable due to memory overflow of the routing tables.

For the good functioning of a network with multiple sinks, it is important that the different sinks transmit the same information in the DIO messages. For most of these parameters, the synchronisation is limited to the start of the construction of the network. However, for the DODAG sequence number, constant synchronisation is needed. This DODAG sequence ID will change if global repair for RPL is executed. Before performing a global repair, the initiating node will contact the other sinks to change the DODAG ID. Because only sinks can execute a global repair, communication between the sinks is sufficient.

4.4.1 Multipoint-to-point traffic

For the collection of data from sensor nodes, only traffic from the sensor to the sink is needed (upwards traffic).

Because each node in the network can reach at least one sink via its preferred parent, no difficulties for multi-sink support arise.

4.4.2 Point-to-multipoint traffic

Since a sink can only directly reach the nodes in its subtree, it is not possible, without modifications and coordination or communication, to reach every node from each sink. An example of this situation is the fact that sink 1 cannot send data to node 4 (Figure 3).

However, all the sinks together can reach every node in the network. In other words if every sink take its responsibility to contact its nodes, there is no problem.

If it is necessary that every sink also has to be able to contact nodes in the subtree of other sinks, this will require communication between the sinks. This can be implemented in different ways. In Section 5.2 the different possibilities will be presented.

4.4.3 Point-to-point traffic

In RPL point-to-point traffic is handled by forwarding the message repeatedly to the preferred parent of each node until, depending on the use of non-storing or storing mode, a common ancestor or the sink is reached. Subsequently, the message will travel downwards until the destination node is reached.

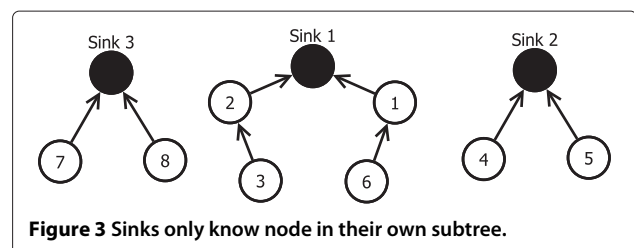


Figure 3 Sinks only know node in their own subtree.

When using multiple sinks, it is possible that a node wants to communicate with a node from another subtree. If this happens the message will be forwarded to the sink of the sending node and subsequently via the correct sink to the destination node (Figure 4). This will also require communication between the sinks, which can be implemented in different ways (*cf.* point-to-multipoint traffic).

Please note that if nodes are in each other neighbourhood, it can be useful to directly send the message to the node (like nodes 4 and 6 in Figure 4). However, the number of cases where this is possible is limited.

5 Implementation of multi-sink support

5.1 Contiki OS

The proposed support for multiple sinks is implemented in the open source Contiki operating system (version 2.6) [17] for evaluation. This operating system is designed to run on devices with limited memory. The embedded COOJA simulator of Contiki makes it possible to emulate adaptations to the system before flashing it to real hardware. Contiki also supports different protocols like the 6lowpan adaptation layer, the IPv6 RPL protocol and the CoAP application layer.

However, the support is implemented in Contiki, the implemented principles are general and can also be implemented in other operating systems like TinyOS. We however prefer to use Contiki, based on the up-to-date implementation of the RPL.

5.2 Sending packets to other subtrees

As mentioned in Section 4.4.2: *Point-to-multipoint traffic*, the communication for forwarding of messages (from a node in the subtree) of one sink to a node in the subtree of another sink can be implemented in different ways.

5.2.1 Forward packet to correct sink

If a sink has to be able to directly forward the packet to the correct sink node, then each of the sinks have to communicate their routing tables over the reliable management link. When using this technique, each node in the network has an entry in the routing table of each sink. For the situation in Figure 5a, this implies that sink 1 stores in its routing table that nodes 4 and 5 are reachable via sink 2.

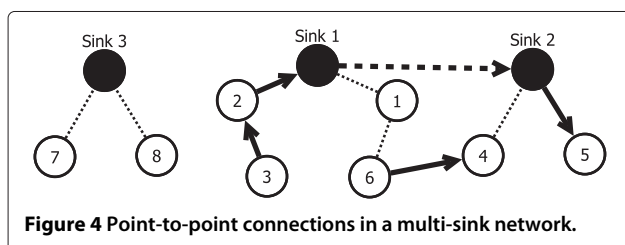


Figure 4 Point-to-point connections in a multi-sink network.

When sink 1 now contacts node 5, the data will flow like illustrated in Figure 5a.

The disadvantage of this technique is that for larger networks, the size of the routing table is not sufficient to store the routing entries of all nodes in the network. In other words this technique, like the usage of a single sink, suffers from the limited storage capabilities of wireless sensor nodes. Another disadvantage is that this solution requires that the different sinks know which sinks exist in the network.

5.2.2 Forward packet to all sink

Another possibility to route messages to a node in another subtree is the forwarding of the messages to all the other sink nodes (eventually via multi-cast) (Figure 5b). This has the advantage that every sink does not need an entry for every node in the network.

The disadvantage is that the packet should be sent to and handled by each sink of the network instead of only the sink which can reach the destination node. Another disadvantage is that this solution still requires that the different sinks know which sinks exist in the network.

5.2.3 Forward packet to central unit

It is also possible to make use of a central unit (Figure 5c), with sufficient memory available to store all routing entries, for the routing of the packets to the different subtrees. The central unit can immediately forward the packet to the correct sink. This technique requires the communication of the routing tables of each sink towards the central unit. This can be preferred if there already exists a central unit to gather the data collected by the different sinks.

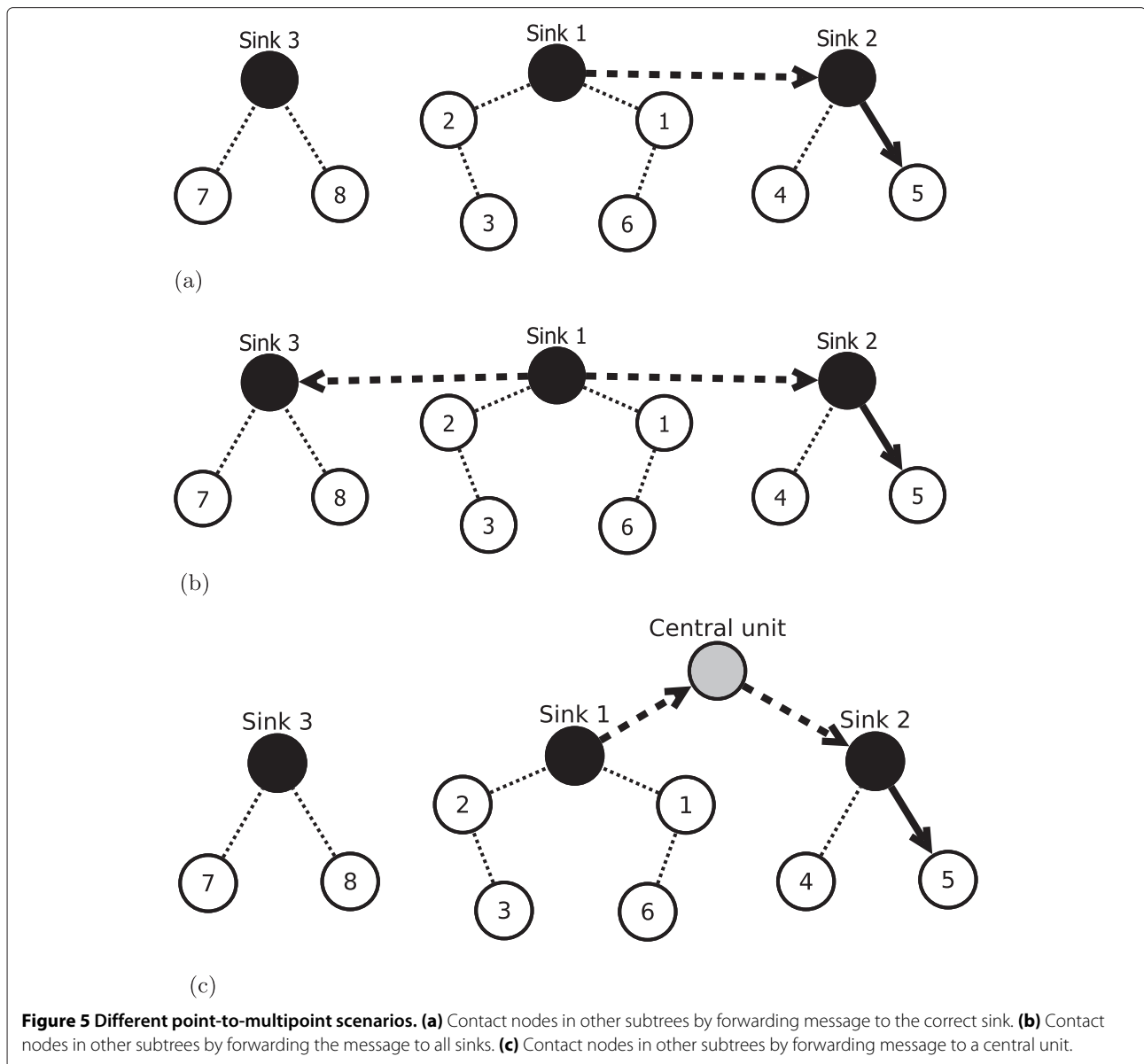
5.2.4 Combination of different techniques

A combination of the previous techniques is also a possibility. An example is the combination of the first (direct to correct sink) and second (sending to all the sinks) techniques. Hereby, the sink directly sends a message to the correct sink if there exists an entry for this node in its routing table (for example due to frequent or previous use). Otherwise the message is sent to all the other sinks.

5.3 Coordination and synchronisation for multi-sink support

5.3.1 Adoption of coordination via a central unit

For the communication between the different sinks, this paper focuses on a centralised approach. This implies that the sinks communicate like in a client-server architecture where one node handles the communication between the different sinks. When a sink wants to signal a change (of a parameter), it will contact the central coordinating node, which will inform the other sinks about the change.



The choice for a centralised approach was made for different reasons. One reason is that when using different sinks, also coordination by a central unit is needed for the collection of the data from the different sinks. A centralised approach has the additional advantage, in comparison to a distributed approach, that only communication with the central unit is needed to contact all the nodes instead of the need to discover all the nodes which have to be contacted. Another advantage is that the routing tables only have to contain the entries for the nodes in the subtree and not the entries of all nodes in the network (like with the forwarding of the packets to the correct sink). A last advantage is that there is no need to send a packet to all other sinks (which otherwise results in an extra overhead) when a packet has to be sent to

a node outside the subtree of a sink. The central unit is called the registrar. In the following subsection, different implementation aspects of the approach are given.

5.3.2 Implementing sink-to-sink communication by adding extra device

For the sake of completeness, we mention that Serial Line Internet Protocol (SLIP) [18] is used for the communication between the sink and the device to which it connected to (e.g. an embedded PC). SLIP is commonly used to encapsulate IP packets for transmission across the serial line of micro-controller devices. SLIP has a low complexity and small overhead. For the communication between the devices (here embedded PCs), any reliable network can be used (e.g. Ethernet).

In Figure 6 a schematic representation of a setup with two sinks and a registrar is made. The sinks are connected to an embedded PC which contains an Ethernet interface. The communication between the sink (sensor node) and the embedded PC makes use of SLIP. For the communication between the embedded PCs (from the different sinks) and the registrar, Ethernet is used.

Contiki already provides support for SLIP communication and includes a *tunslip* tool [19] which make it possible to communicate with devices using SLIP. The tool constructs a SLIP tunnel between a physical serial interface and a virtual network adaptor. By using *tunslip* the communication between the sink and the embedded PC is facilitated. If we refer to Figure 6, *tunslip* can be used to bridge the traffic from the serial port to the Ethernet adaptor.

5.4 Anycast addressing

Since the sensor nodes are not aware of the existence of multiple sinks, they will send their information to the virtual sink. It is the task of the different sinks to intercept this information and handle it themselves (as the virtual sink does not physically exist). An easy solution for this is to assign an identical anycast address [20] to all the sinks and use this as the address of the virtual sink. When a sensor node wants to contact the virtual sink, it will use this anycast address which will make sure that one of the different sinks will process the data.

In Contiki, anycast packets are not forwarded to the application layer. However, this has as consequence that when the packet successfully reached the sink, the application was not informed. Therefore, we adapted Contiki to solve this problem by also checking on anycast addresses and prevent dropping the anycast packets.

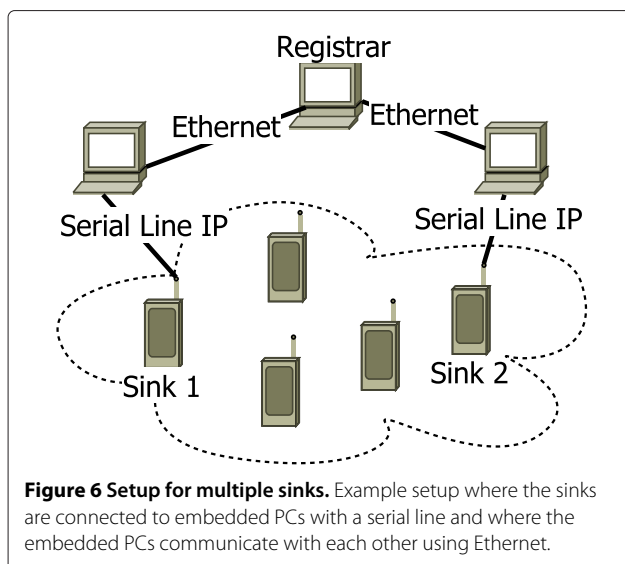


Figure 6 Setup for multiple sinks. Example setup where the sinks are connected to embedded PCs with a serial line and where the embedded PCs communicate with each other using Ethernet.

5.5 Communication applications

To implement the communication between the sinks and the registrar in a client-server approach, two applications were implemented. The first application (server/registrar) will coordinate the registration of the different sinks of the DODAG and collect different parameters of the DODAG (like DODAGID and DODAG sequence number). If a sink signals a parameter change, the application will inform the different sinks. The parameter changes will be signalled to the registrar by another application, which also will be informed by the registrar if a parameter change was signalled by another node. This application will be executed by each sink in the network. Currently, only the DODAG sequence number will change during the lifetime of the network.

5.5.1 Register a sink

When a sink joins a RPL network, it will first register with the registrar so it knows which sinks are part of the network. This enables the registrar to inform all sinks in the network about the change of parameters. When a sink registers to the registrar, also the current parameters are communicated to the joining sink. Afterwards the newly registered sink can start to transmit DIO messages.

Figure 7 illustrates the joining of two sink nodes. When node 1 is booted, it will inform the registrar it wants to join the DODAG and asks for the information concerning the DODAG. Because the sink is the first node for the DODAG, it will inform the sink that it is the first sink in the network and permits the sink to start its own DODAG. The sink will start transmitting of DIO messages and inform the registrar about the chosen parameters.

When a second sink registers, the registrar knows the parameters and will communicate them to the new sink. Thereafter, the new sink will initialise the parameters correctly and starts transmitting DIO messages.

Depending on the used objective function, a global repair can be initiated every time a new sink joins the DODAG to force the nodes in the network to construct a completely new DODAG. In this paper this technique was not used because the chosen objective function (MRHOF) choose the preferred parent based on the rank, which means a node will automatically switch when a node closer to the sink is detected.

5.5.2 Execute a global repair

As mentioned above, when a global repair is executed (due to an inconsistency), all the joined sink nodes have to be informed about it. In Figure 8 the communication needed to inform all nodes about an imminent global repair is illustrated.

The sink that will execute a global repair will inform the registrar about it. As a result of the execution of a global repair, the version number of the DODAG has to

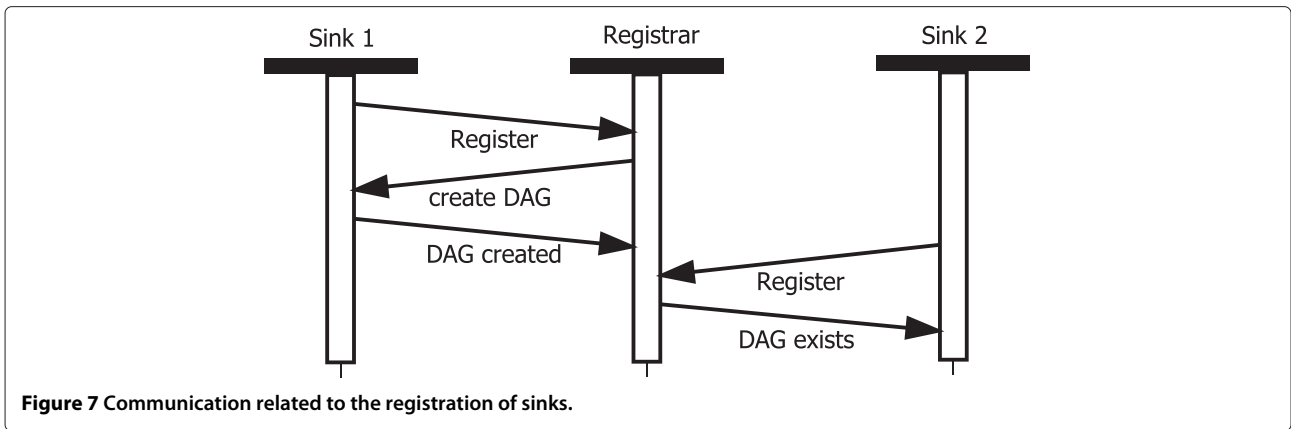


Figure 7 Communication related to the registration of sinks.

be increased. When the registrar gets the request for the execution of a global repair, it will inform the other sinks about the new version number. Please remark that the sinks have to wait until they get permission to send out the new version number. Once all the sinks are informed about the new version number, the registrar will grant permission to all sinks to use the new version number (and execute the global repair). This permission is to prevent that a sink receives a DIO with the new version number before it is informed about the new version number because this would result in a new global repair. If however a sink receives a DIO with the new version number before it gets permission to use the new version number (but after he is informed about it), it will start using the new version number, without initiating a new global repair, because this indicates that another sink has already got permission to use the new version number.

5.6 Routing when using multiple sinks

5.6.1 Routing from node to a sink

For the routing from a node in the tree of a sink to that sink, the packets will be routed like described in the RFC describing RPL. The upward routing principle is based on the selection of preferred parent for each node in the network. This selected parent will act as the next hop for

the multi-hop path towards the sink. The preferred parent will be selected based on the objective function used to construct the DODAG.

5.6.2 Routing inside the subtree of a sink

For routing from one node towards another node inside the tree of one sink, the same techniques are also used which are specified in the RFC. In the non-storing case of RPL, a packet will always travel until it reaches the DODAG root which sends it to the destination node, because the nodes in the network do not store the nodes in their subtree. In the storing case, nodes in the network store the nodes in their subtree in their routing table. When a packet, on its route towards the DODAG root arrives at a common ancestor of the sender and the destination, it is redirected down towards the destination. When using multiple sinks, the different sinks will also function as a DODAG root which knows the nodes in its subtree.

5.6.3 Routing from one subtree to another subtree

When in our case nodes with different sinks want to exchange messages, the messages will firstly be sent to the sink of the sender via the different preferred parents. The sink has no routing entry in its table, which indicates that the destination node is not situated in the subtree of

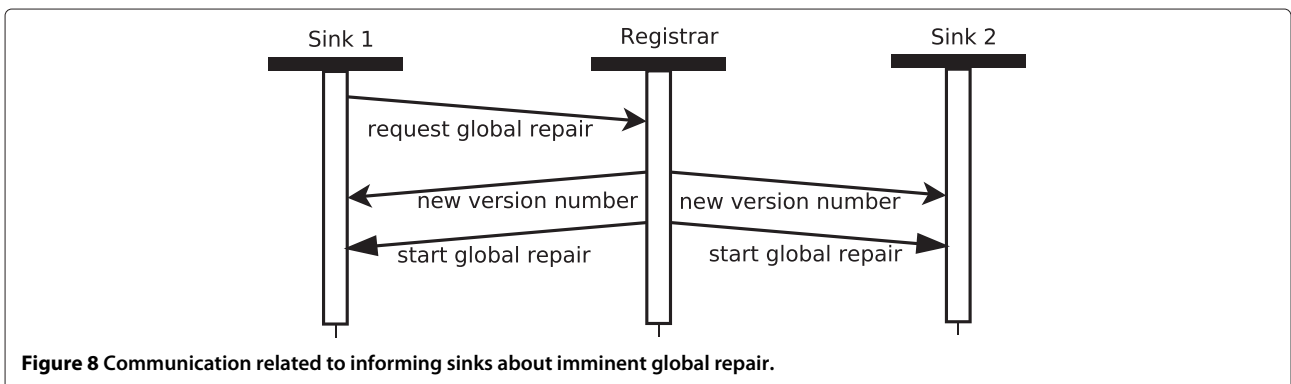


Figure 8 Communication related to informing sinks about imminent global repair.

the sink. Then the sink will send the message to registrar, which will send the message to the sink of the destination node. Thereafter, this sink will send the message to the destination node.

The registrar needs an overview of the positions of all the nodes in the DODAG, and more specifically which sink serves which node. Therefore, the DAO messages sent by the nodes in the network, to indicate the downward route to the node, have to be forwarded towards the registrar.

6 Multi-sink simulations

To evaluate the performance of the implemented support for using multiple sinks in a wireless sensor network with RPL routing, some tests were performed with the COOJA simulator [21].

6.1 The performance test

All sensor nodes (except the sinks) run an application that periodically (each 50 to 70 s) sends a data packet to the closest sink. The data packet has an ID (to determine the number of lost packets) and contains information about the node (rank, energy consumption, preferred parent, ETX value to the preferred parent,...). The information in the data packet makes it possible to determine the average

energy consumption, the structure of the DODAG, the average number of hops,....

6.2 Simulation setup

For the simulations, the COOJA simulator was used. This simulator is integrated in the Contiki environment and can run the same executables as these for real-life nodes. As 'link failure model' the 'Unit Disk Graph Model (UDGM)' was used. This model is configured with the transmission distance (maximal distance where a packet still can be received) and interference distance (maximal distance of the collision domain). The environment also uses the transmit (percentage of packets that are successfully transmitted) and receive ratio (percentage of packets that are successfully received at the maximum transmission range). In the UDGM the probability that a packet is received by a node will decrease exponential from the transmit ratio (next to the sender) to the receive ratio (at the maximum transmission range). The probability of receiving a packet can be described by $TX \times (1 - (D^2/R^2)) \times (1 - RX)$ with TX the transmit ratio, RX the receive ratio, D the distance between sender and receiver and R the transmit range.

The setup (Figure 9) consists of one registrar (node 1 not depicted), four sink nodes (nodes 2 to 5) and thirty nodes

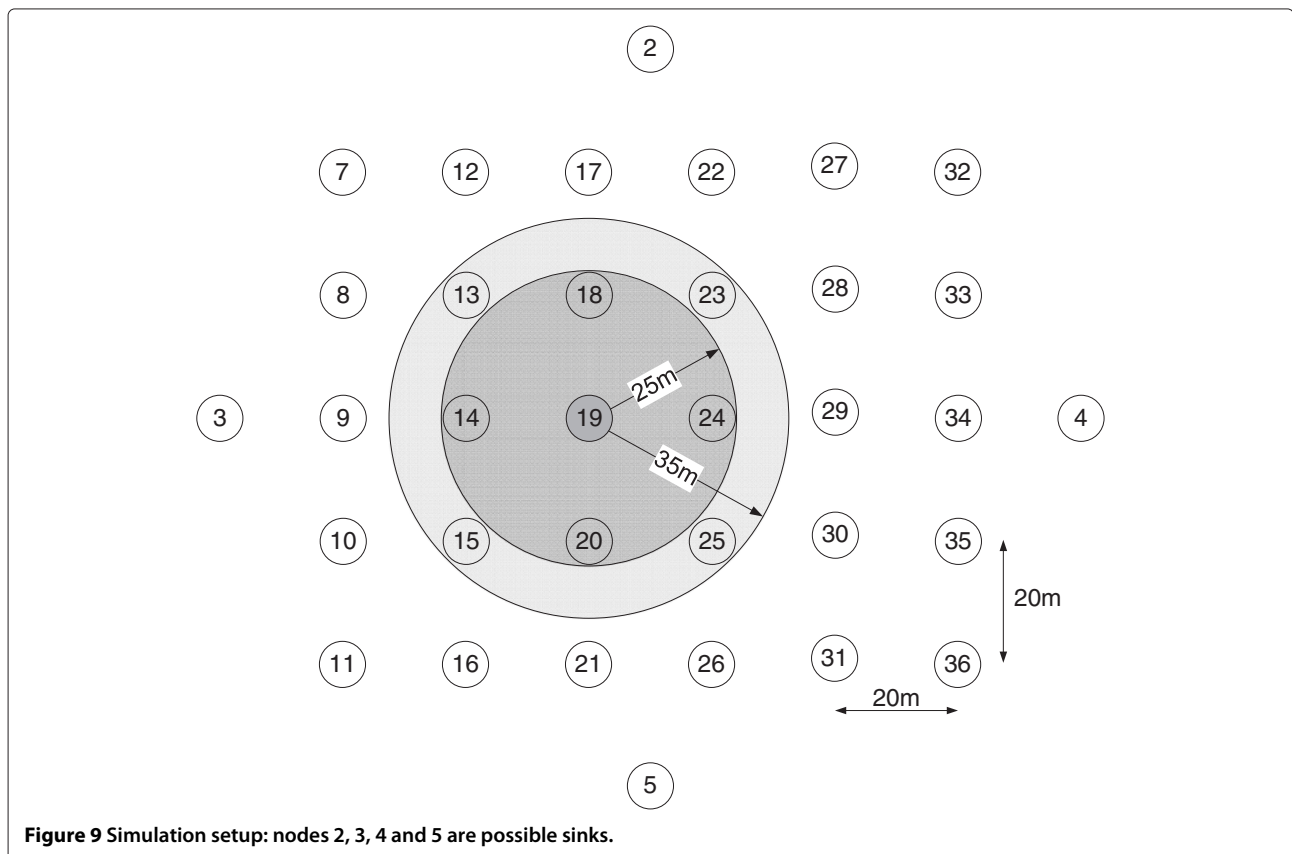


Figure 9 Simulation setup: nodes 2, 3, 4 and 5 are possible sinks.

evenly placed (20 m between two nodes) in a grid structure (nodes 7 to 36) which periodically send data packets. Concerning the positioning of sink nodes, a lot of research papers already exist as proven in Section 2. As depicted in Figure 9, we assume that the sinks nodes are placed at the middle of each side of the grid structure. This choice is made to obtain an evenly distribution of the nodes over the four sinks. Possibly by positioning the sinks on an even better place, the performance can even be increased.

The transmission range of the nodes is 25 m and the interference range is 35 m. For the MAC protocol the CSMA protocol was used, and for the RDC protocol the Contiki-MAC protocol was selected. To ensure the packet delivery to the sink, the RDC protocol for the sink was disabled. This prevents the radio of the sink being switched off. The MAC protocol is allowed to send a packet for the second time in case no ACK was received the first time.

The simulated platform is the Sky-platform. These Tmote Sky motes have a nominal current consumption [22] of 21.8 mA when receiving, 19.5 mA when transmitting and 1.8 mA when the radio is off. The simulations were performed during 60 min. To register the energy consumption of each node, the Contiki Powertrace tool was used. The average number of hops is calculated by taking the average, for each node, of the number of hops a packet had crossed when it arrives at the sink.

The simulated environment is rather small for the usage of four sink nodes, but the goal of these tests is to demonstrate the behaviour of the implemented support.

To test the implemented support for multiple sinks and the influence of the use of multiple sinks on the performance of the network, different performance tests were executed with different numbers of sinks. The adding of extra sinks was based on the node ID. This means that at

first, only node 2 was activated, next nodes 2 and 3, and so on.

6.3 Results

6.3.1 Influence on the number of hops

In this subsection we want to illustrate the influence of distance (in number of hops) to the nearest sink on packet loss and total consumed energy. In the test to illustrate this influence, one sink was used.

In Figure 10 the distribution of the nodes related to the average number of hops to the sink is illustrated. In Figure 11 the chance of packet loss related to the average number of hops per node for the path between the node and the sink is represented. For these tests the maximal number of transmissions for a packet has been set to 1. For the receive ratio, a percentage of 50% has been selected, which results in about 32% packet loss for the links between two nodes which are positioned 20 m from each other.

Figure 11 illustrates that when the number of hops increases, also the probability of packet loss increases. The results show that if more than 3 hops are needed, more than 50% of the packets get lost. This illustrates that a reduction of the average number of hops can contribute to less packet loss.

Another evaluation that is made is the average energy consumption related to the average number of hops (to the sink). In a first series of tests, no interference was assumed (receive ratio of 100%). Figure 12 illustrates that nodes closer to the sink typically consume more energy compared to nodes farther (more hops) from the sink. This can be explained by the fact that the nodes in the neighbourhood of the sink are responsible for the forwarding of data from the lower nodes in the subtree. So

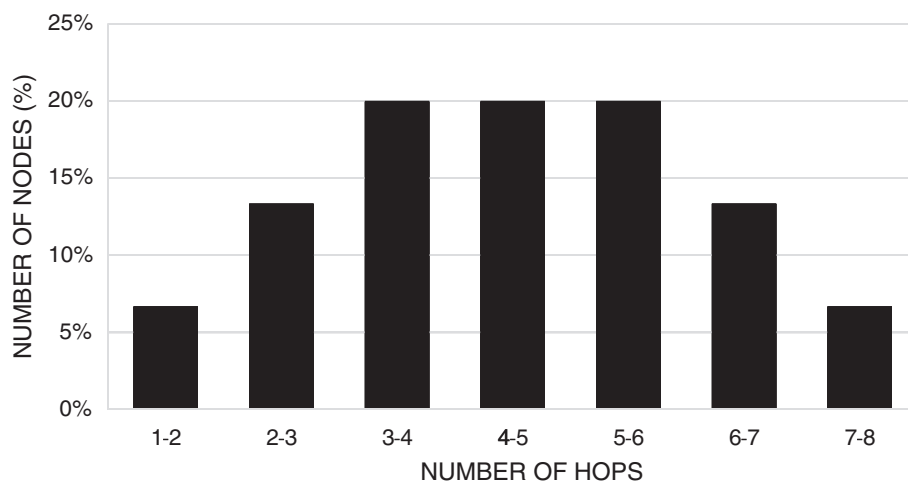


Figure 10 Distribution of nodes related to the average number of hops per node. Distribution of nodes related to the average number of hops per node for the path between a node and the sink (with 30 nodes and one sink in a 6 × 5 grid).

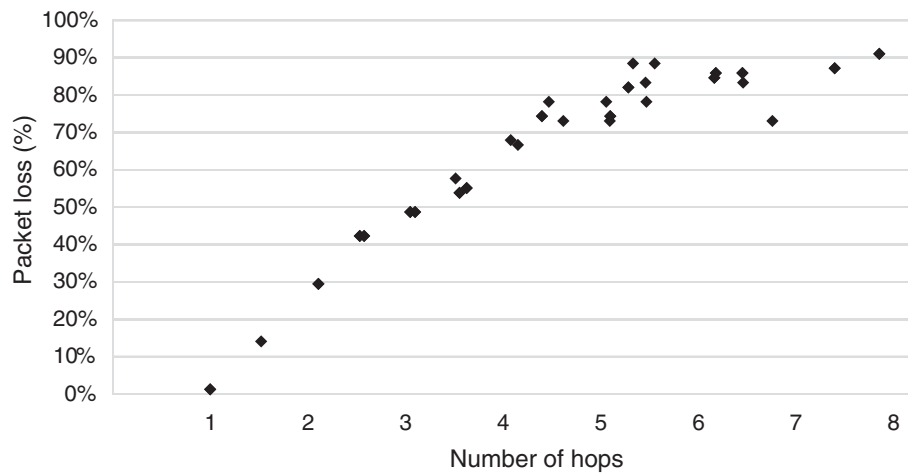


Figure 11 Percentage of total packet loss related to the average number of hops per node. Percentage of total packet loss related to the average number of hops per node for the path between a node and the sink (with 30 nodes and one sink in a 6×5 grid).

a smaller DODAG or better distribution of nodes over the DODAG results in less descendants and lower energy consumption.

A packet receive ratio of 100% is not realistic in a wireless sensor network environment due to the lossy characteristics of the environment. Therefore, we reduced the receive ratio to 50% to simulate a more realistic environment. These settings are comparable to the behaviour experienced during the tests on a real-life office testbed. If we compare the results for 100% receive ratio with the results in Figure 13, a more scattered energy consumption is observed. This scattered energy consumption can be explained by the increase in packet loss resulting in more control traffic which increases the energy consumption. We still can conclude that nodes at the bottom of the DODAG tree (in this case for nodes with more than 6

hops) consume less energy because they are less involved in the forwarding of packets from nodes in their subtree.

However, all nodes with the highest energy consumption are not situated any more in the neighbourhood of the sink, but also in the middle of the DODAG tree. This can be explained by the increase in control packets to solve, for example, a loop which is detected (local repair).

Also, the ratio between application data and control traffic influences this graph. In the tests the application data is limited to a packet per minute per node.

6.3.2 Influence of multiple sinks

In this section a closer look to the effect of using multiple sinks is taken. This effect is evaluated based on simulations with a receive ratio of 50% and maximal one retransmission.

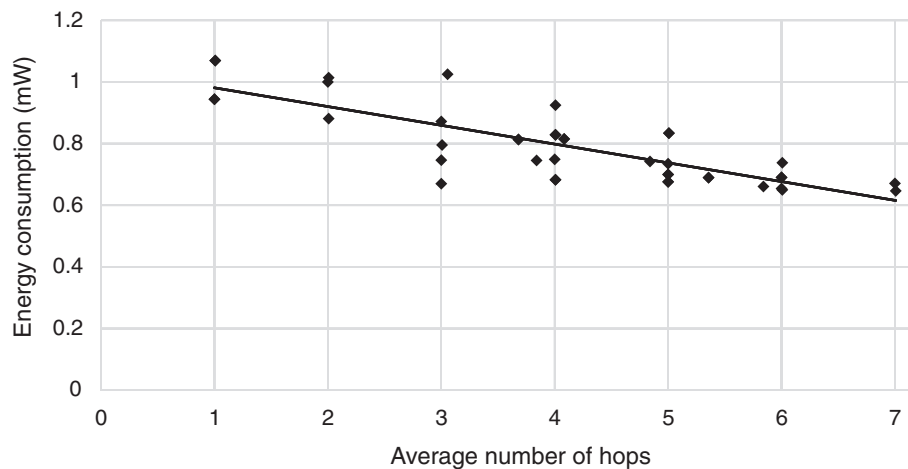


Figure 12 Average energy consumption related to the average distance to the sink in number of hops. Average energy consumption related to the average distance to the sink in number of hops (receive ratio = 100%; linear trendline; 30 nodes and one sink in a 6×5 grid).

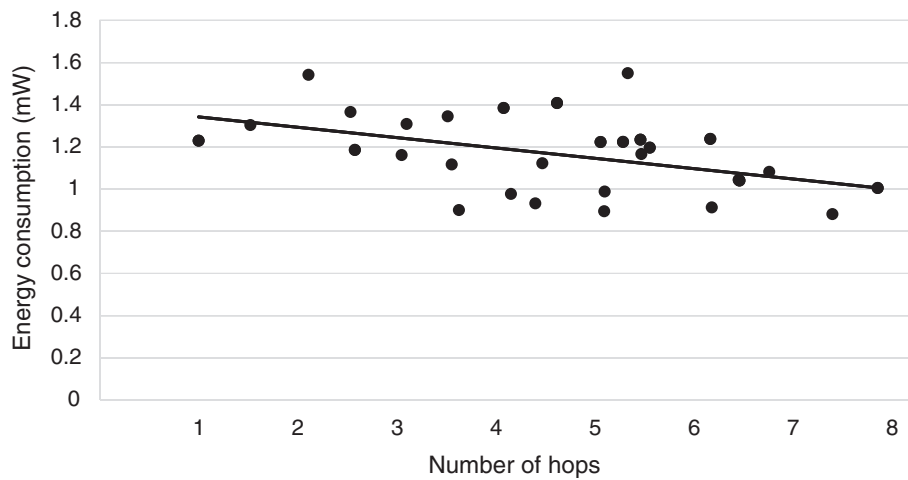


Figure 13 Average energy consumption related to the average distance to the sink in number of hops. Average energy consumption related to the average distance to the sink in number of hops (receive ratio = 50%; linear trendline; 30 nodes and one sink in a 6 × 5 grid).

DODAG structure Figures 14 and 15 provide a view on how packets are routed from the different nodes in the network towards the sink, for respectively one and four sink nodes. Both figures show that the behaviour is as expected. In Figure 14 it is illustrated that the traffic from all nodes in the network is routed over only two nodes (nodes 17 and 22). Instead of one big DODAG, the construction of the different smaller DODAGs is illustrated in Figure 15. However, if the virtual sink would be added in Figure 15 also one big DODAG would be obtained. However, this DODAG would differ from the one in

Figure 14 because it would be wider instead of deeper. This means that the use of multiple sinks can lead to a reduction of the average distance, in hops, from a node to the sink and a more equal distribution of the traffic over the network. If the number of hops to travel decreases, then the packet loss also decreases (*cf.* Figure 11). The construction of a wider DODAG also has as advantage that in a DODAG, with a collection purpose, the different nodes have less descendants which induces a lower forwarding packet load. This also benefits the energy consumption in these nodes.

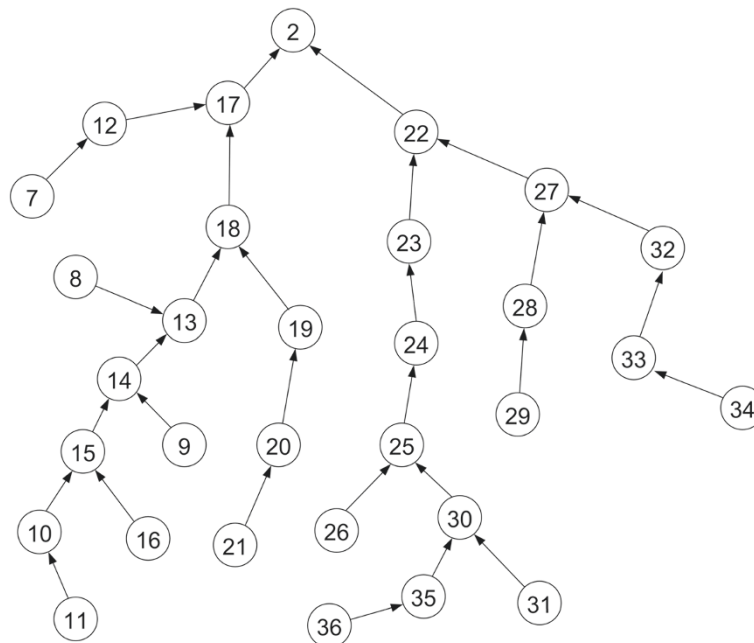


Figure 14 DODAG constructed by one sink.

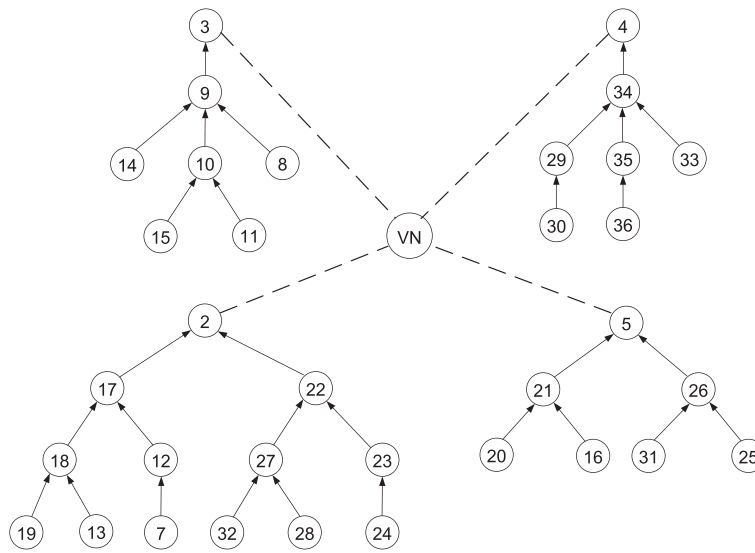


Figure 15 DODAG constructed by four sinks interconnected by a virtual node.

Number of hops The reduction of the average number of hops to the nearest sink is also illustrated in Figure 16 where the average number of hops is compared to the number of sinks. This figure also illustrates the minimal and maximal average number of hops per node.

Beside a reduction of the average number of hops, also the reduction of the maximal average number of hops can be perceived when more sinks are added to the network. The exact structure of the graph, and the advantage of adding extra sinks, highly depends on the exact placement of the sinks. In this setup the average number of hops can be reduced with 46.8% if, instead of one, four sinks are used. The maximal average number of hops is even reduced with more than 50%.

Packet loss In Figure 17 the chance of packet loss is compared to the number of sinks used. The graph also illustrates the minimal and maximal packet loss for each node. In addition, the graph illustrates the reduction of the maximal and the average packet loss if the number of sinks in the network is increased.

If we compare Figures 16 and 17, they have a similar behaviour, which is illustrated by joining both figures in Figure 18. This effect illustrates again the relation that each hop that needs to be taken will increase the chances of packet loss as shown in Figure 11.

Since the sinks are placed at four completely different positions, collisions are less likely to happen when all four are activated. When using only one sink, all the data will

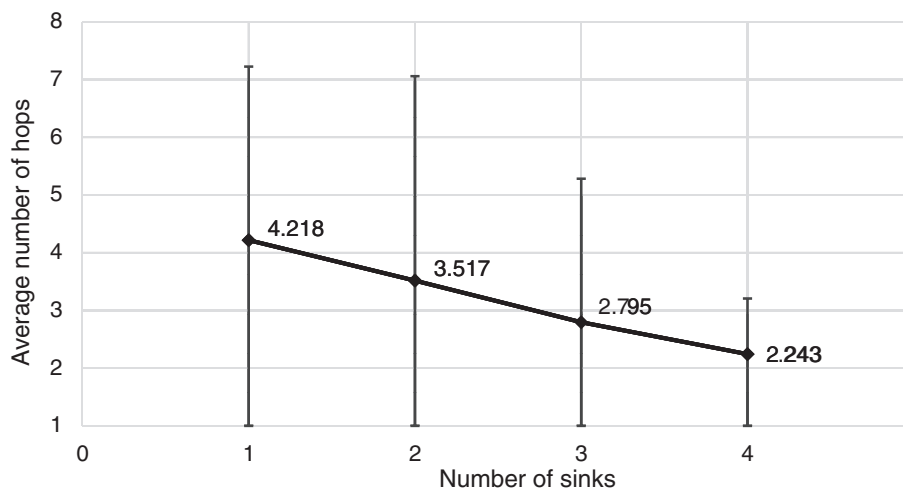


Figure 16 Average number of hops compared to the number of sinks. Average number of hops compared to the number of sinks (maximum and minimum number of hops are represented by the error bars) (30 nodes in a 6 × 5 grid).

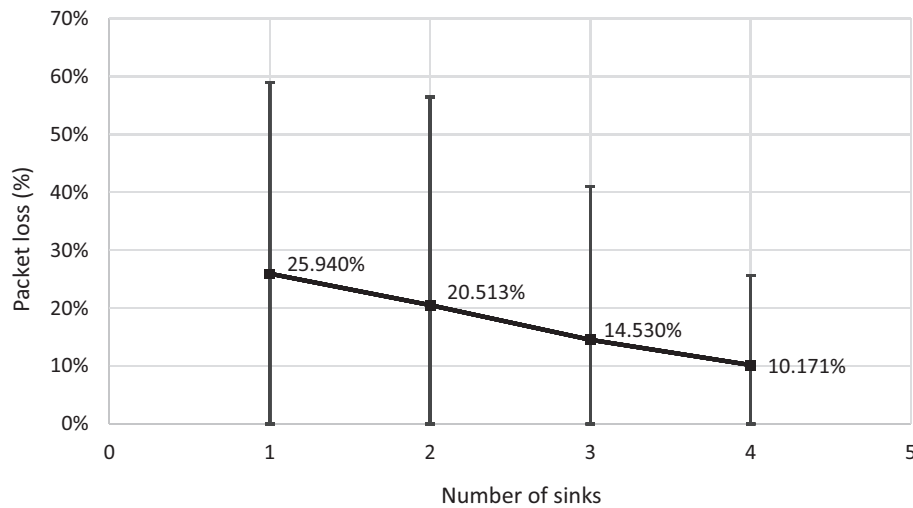


Figure 17 Average packet loss compared with number of sinks. Average packet loss compared with number of sinks (maximum and minimum packet loss are represented by the error bars) (30 nodes in a 6×5 grid).

flow in the same direction (e.g. towards the top of the network), and this will increase the chances of packet loss due to collision. By adding more sinks, the data will flow in different directions and the chances of packet collision will be reduced. This is another reason why the packet loss decreases as the number of sinks increases.

Energy consumption Finally, the influence of the use of multiple sinks on the energy usage of the network nodes is evaluated. The results of these tests are depicted in Figure 19. In the graph also the minimal and maximal energy consumption is illustrated.

Just like the packet loss, the average (and the maximum) energy consumption also decreases as the number of sinks increases. This is also the result of the

reduction in the average number of hops because each node is now positioned closer to the sink. Because the receiving and sending of packets over the radio consume the main part of the energy of a node, the reduction of the number of hops will lead to less energy consumed to transmit a packet, over different hops, to the sink. In other means this reduces the average energy usage.

Maybe more important is the reduction of the maximal energy consumption by a node. If we assume that every node in the network has the same energy capacity, the reduction of the maximal consumed energy by a node will extend the lifetime of the network (first energy failure of a node). Because the nodes in the direct neighbourhood of the sink belong to the biggest energy consumers, the

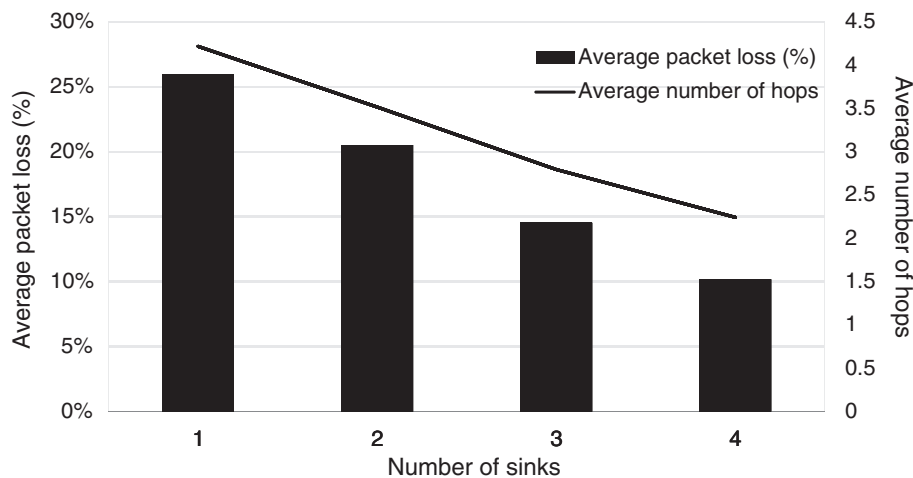


Figure 18 Packet loss and average number of hops compared to number of sinks used. Packet loss and average number of hops compared to number of sinks used (30 nodes in a 6×5 grid).

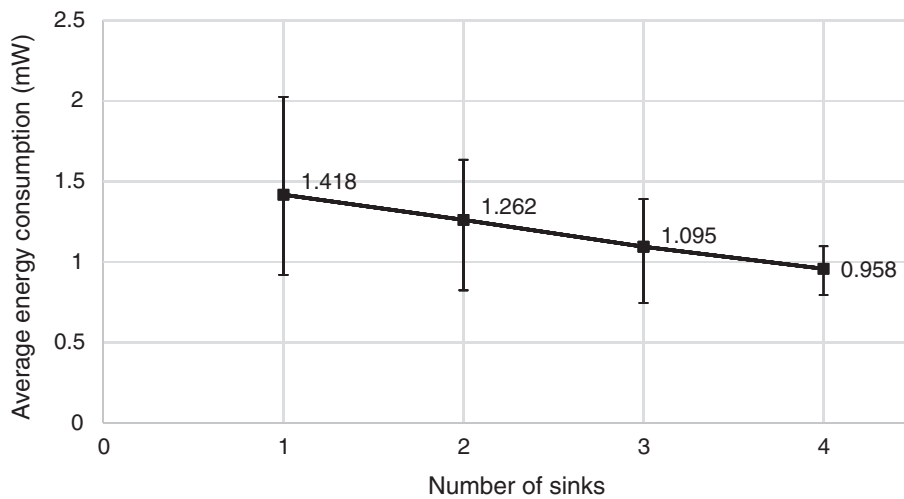


Figure 19 Average energy consumption compared to the number of sinks. Average energy consumption compared to the number of sinks (maximum and minimum energy consumption represented by the error bars) (with 30 nodes in a 6 × 5 grid).

failure of a few nodes can lead to the inability of nodes to communicate with the sink, as illustrated in [23].

7 Experimental evaluation: iMinds wiLab.t office testbed

To validate the results from the performed simulations, the same tests, although on a different topology, were performed on a real-life testbed (iMinds wiLab.t) [24]. The setup and obtained results are discussed in this section. For these real-life tests, the maximal number of retransmissions was increased to 3. The number of retransmissions was increased because some links in the testbed were less reliable due to obstacles, like walls in the office testbed.

For the wireless communication in these tests, the IEEE 802.15.4 interface with channel 11 and a transmit power of -15 dBm (level 7) was used. These settings were chosen to obtain a multi-hop network.

7.1 Testbed setup

The topology and selected nodes from the wiLab.t testbed are displayed in Figure 20. The wiLab.t testbed [25] is

an office environment which consists of Tmote sky sensor nodes connected to an embedded PC installed on almost 200 locations on three floors. These nodes are the same as used for the simulations. The embedded PCs are interconnected via a switch over Ethernet.

As mentioned earlier, the different sinks and the registrar will communicate by making use of SLIP (for communication between the sensor node and the embedded PC) and IP tunnels (for communication between embedded PCs). The results were collected on the embedded PC on which the registrar was connected (node 91) and via a VPN tunnel transferred to the collect-view application. This setup is depicted in Figure 21.

7.2 Results

7.2.1 Influence of the number of hops

First the effect of the number of hops will be evaluated while using one sink (only sink 1 from Figure 20).

Packet loss In Figure 22 the effect of the number of hops on the packet loss is depicted. If these results are compared to the simulation results from Figure 11, the fixed

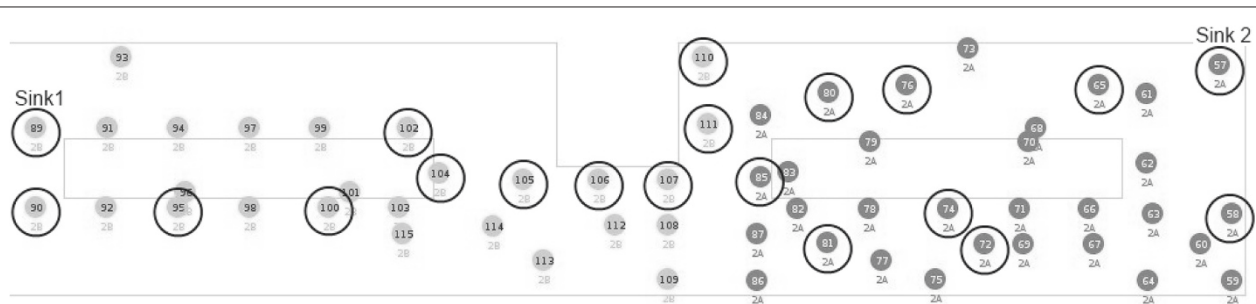


Figure 20 Topology for real-life tests. Nodes 89 and 57 are respectively sinks 1 and 2. The other nodes are all senders (18 in total).

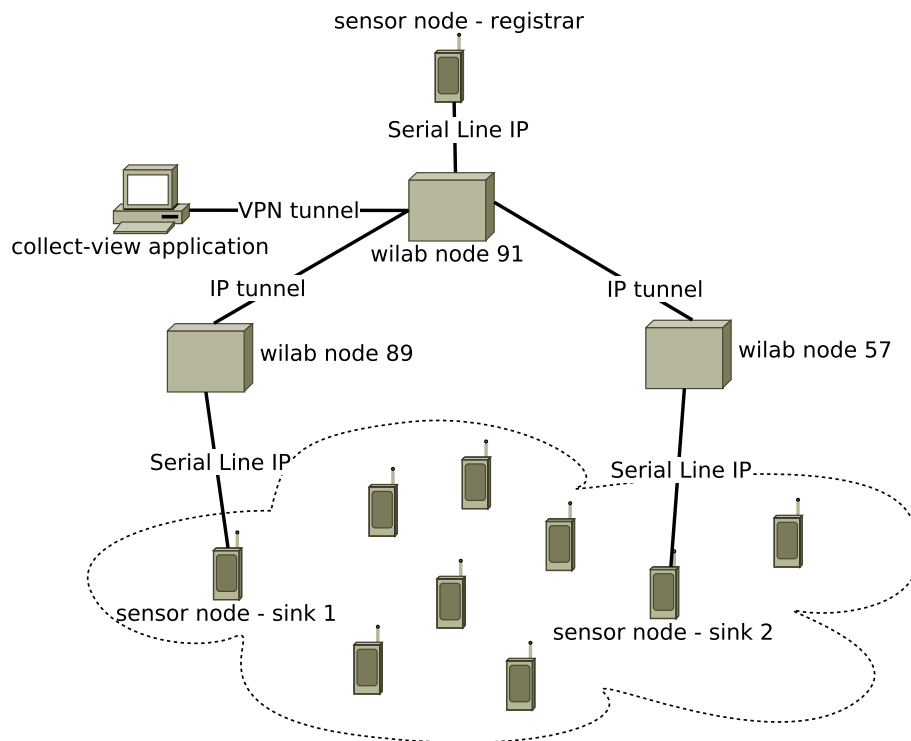


Figure 21 Real-life test setup. Communication between sinks and the registrar (via SLIP and IP tunnels) and between the collector and collect-view application (runs on a machine outside the wiLab.t network) via a VPN-tunnel).

relation between the number of hops and the packet loss is less strict. For nodes with an average number of hops lower than 2 from the sink, there was no packet loss detected. This is probably the effect of choosing for three retransmissions. But for an average of more than 3 hops, the packet loss increases fast. For an average of 4 to 5 hops, a packet loss of 90% is experienced. This bad packet

delivery is possibly caused by the bad link quality of some intermediate nodes which serve as parent for some other nodes. By adding additional sink nodes, the average distance from each node to the sink can be reduced, which, in accordance to these conclusions, can lead to a decrease in the average packet loss. The addition of an extra sink node also delivers alternatives to nodes with a bad link

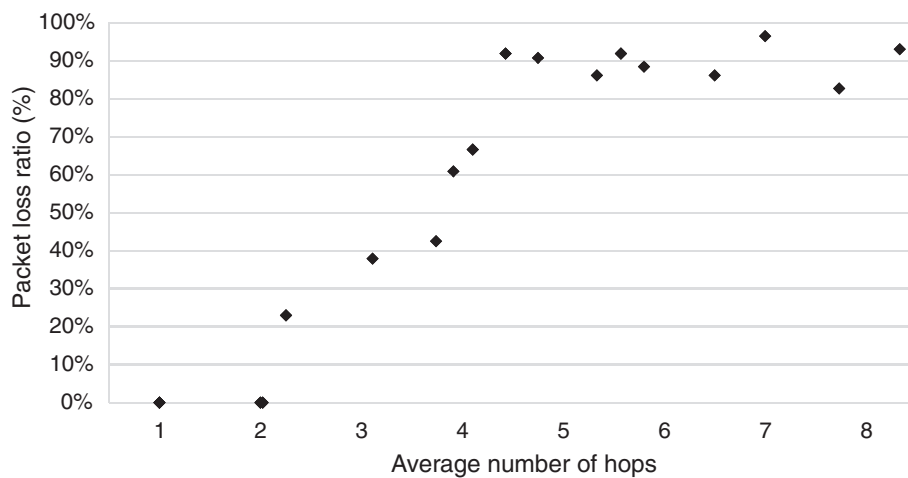


Figure 22 Percentage of packets that get lost in function of the average number of hops. Percentage of packets that get lost in function of the average number of hops to the sink (with one sink) on the wiLab.t testbed.

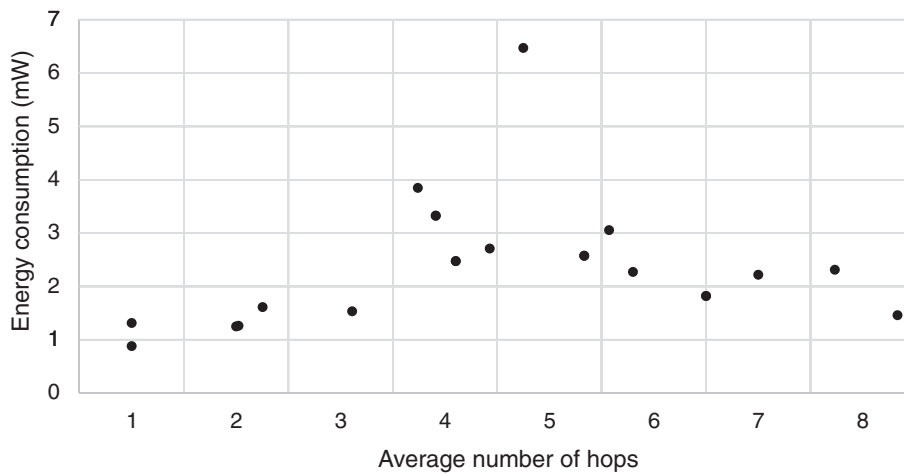


Figure 23 Average energy usage in function of the average number of hops to the sink. Average energy usage in function of the average number of hops to the sink (with one sink) on the wiLab.t testbed.

quality to their preferred parent. Another solution to increase the successful delivery of packets is to increase the number of retransmits. This worse packet loss can also be explained by the interference by other wireless equipment in the office environment where the testbed is installed.

Nevertheless, with the increase in the number of retransmissions, there is still a significant increase of the packet loss ratio once a node sends packets to the sink over more than 4 or 5 hops.

Energy usage In Figure 23 the effects of the number of hops compared to the energy consumption is illustrated. In comparison to the simulation results in Figure 13, the behaviour is completely different. For the results of the test on the testbed, the high consuming nodes are located on average on 4 to 5 hops from the sink. This behaviour

can be explained due to the difference in link conditions which are, compared to the simulation model, not the same for all nodes. As with the simulation tests (with packet loss) also here the local recovery mechanism will be responsible for an increase in the power consumption when the repair mechanism for detected loop is initiated.

In order to evaluate the interference effects on the links to the preferred parents on the different node positions, the average ETX to the preferred parent of a node is compared to the number of hops to the sink and displayed in Figure 24. The figure illustrates that the node with the highest energy consumption in Figure 23 also has a high ETX value. A higher average energy consumption for a node with a bad link to the parent (high ETX value) is logical because the ETX value represents the average number of times a packet has to be transmitted before it successfully reaches the parent. This implies a higher number of

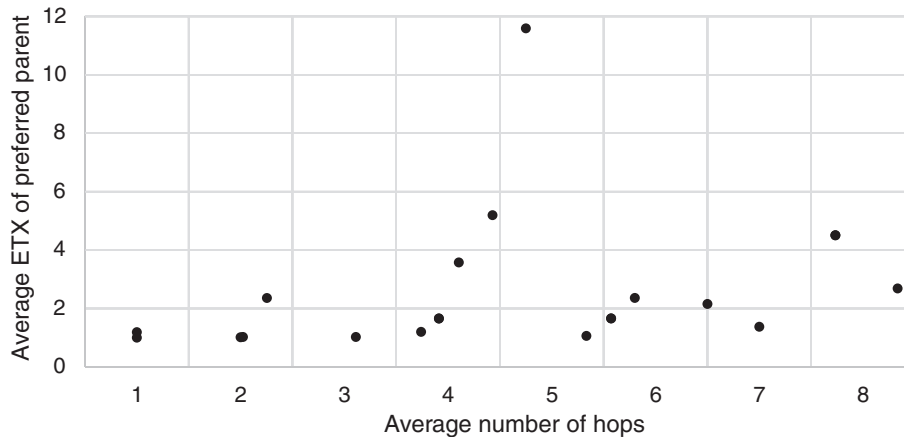


Figure 24 Average ETX value of the path between preferred parent and node versus number of hops. Average ETX value from the path to the preferred parent of a node related to the average number of hops to the sink (with one sink) on the wiLab.t testbed.

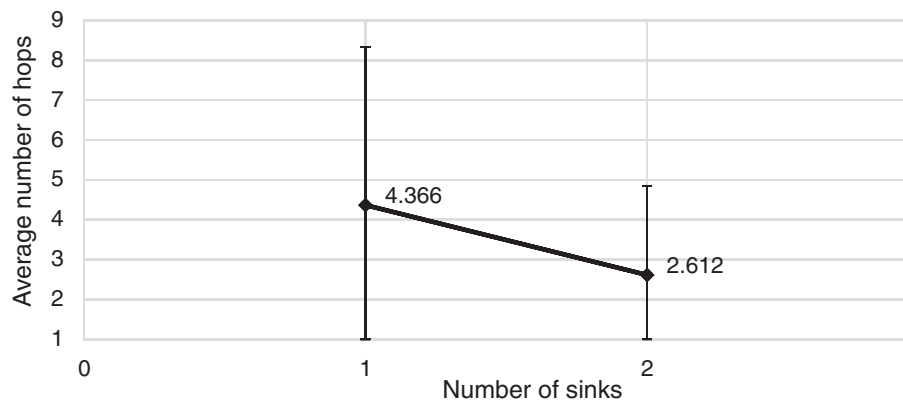


Figure 25 Average number of hops related to the number of sinks on wiLab.t testbed.

retransmission and a higher energy consumption for the successful reception of the packet.

7.2.2 Influence of the usage of multiple sinks

Number of hops If an extra sink is used (beside sink 1 also sink 2 from the topology depicted in Figure 20), the overall average number of hops to the sink has to decrease. In Figure 25 the expected decrease of average maximal number of hops and average number of hops is depicted. The decrease for both parameters is approximately 40% when the number of sinks increases from one to two sinks.

Packet loss The effect of the reduction of the average number of hops on the packet loss is presented in Figure 26. Although the number of hops decreased with 40% when an extra sink was added, the average packet loss decreased with 60%. This stronger reduction is related to high packet loss of more than 90% when using one sink for nodes with more than 4 hops. This is illustrated by the reduction of the maximal packet loss from more than 90% to less than 70%.

The relationship between the reduction of number of hops compared to the packet loss percentage for an increase of number of sinks is also illustrated in Figure 27.

Energy usage Adding an extra sink to the network during the testbed experiments resulted also in a reduction of the average energy consumption (Figure 28). In Figure 28 the strong reduction of the maximal energy usage is remarkable. This strong reduction can be explained by the increase in possible parents for the one node with only bad link qualities to its parents. The addition of the extra neighbour on the other side of the building gave the choice to select parents in the other direction of the building. In this set of possible extra parents, there was a parent to which the node had a link with a better quality. The conclusions concerning the increase in lifetime of the network due to the reduction of maximal energy consumption, from the simulation, are also applicable for the tests on the testbed. In this case the network lifetime can even be doubled due to a reduction of more than 50% of the maximal consumed energy.

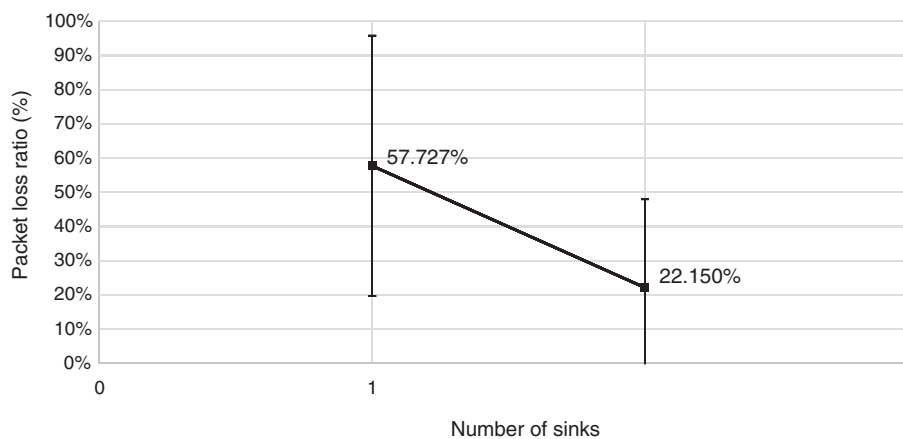


Figure 26 Packet loss ratio related to the number of sinks on the wiLab.t testbed.

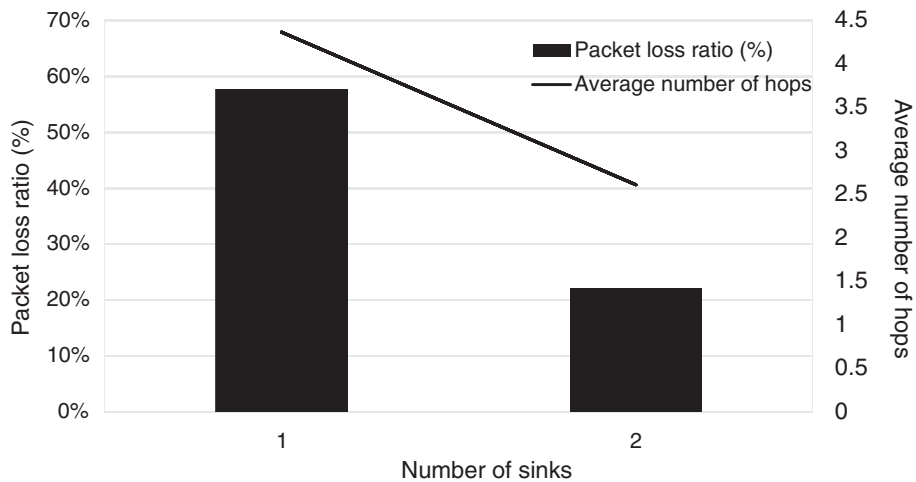


Figure 27 Packet loss and average number of hops in function of the number of sinks. Packet loss (bars) and average number of hops (line) in function of the number of sinks on the wiLab.t testbed.

8 Effect on point-to-point connections

The splitting of the DODAG tree of Figure 14 which results in the tree of Figure 15 can also have some disadvantages for the point-to-point communication between two nodes in the network. Because the network is splitting up in smaller trees, the path between neighbouring nodes is sometimes enlarged. An example of such a situation is the path between nodes 15 and 16. In the DODAG with one sink, these nodes are direct neighbours. However, when more sinks are used, the nodes choose a different sink and the path is enlarged from one 1 to 7 hops. Together with path enlargements, also path reductions are possible. An example of such a situation is the path between nodes 21 and 26. This path is reduced from 10 to 2 hops.

The enlargement or reduction of the path between two nodes is also possible in a DODAG with one sink. The structure of a DODAG can change over time. An example

is that after a global repair, node 13 switches its preferred parent from node 18 to node 12. This will reduce the path length between nodes 12 and 13 from 3 hops to 1 hop and will increase the path length between nodes 13 and node 18 from 1 hop to 3 hops.

If we now compare the hop count for all the point-to-point connections between all the nodes, we notice that the maximum hop count is reduced from 14 to 8 hops. This is achieved because the maximum path length between a leaf node and the (virtual) sink is reduced from 7 to 4 hops. The average hop count is reduced from 6.06 to 5.4 hops.

For both DODAG structures the median is 6 hops, but the standard deviation decreases from 3.00 (for one sink) to 1.83 (for four sinks).

The results are however related to the storing case of RPL. If however the network is configured in non-storing mode, all the communication between the nodes is sent

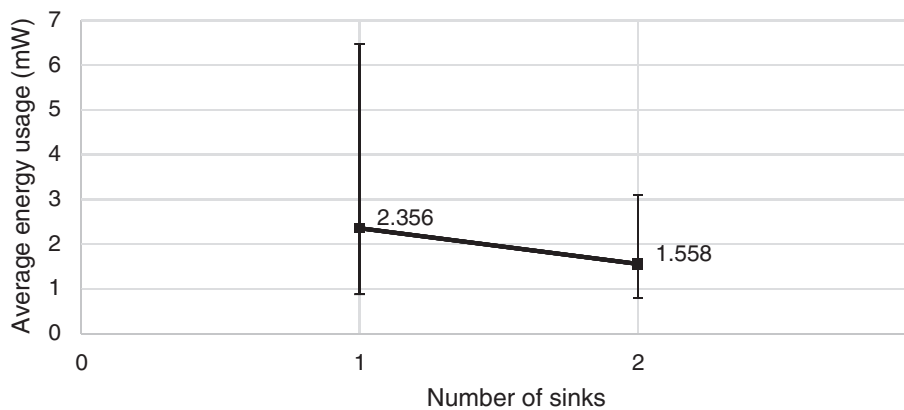


Figure 28 Average energy consumption in function of the number of sinks. Average energy consumption in function of the number of sinks (maximum and minimum energy consumption are represented by the error bars) on the wiLab.t testbed.

via the DODAG root, because the nodes in the network do not store information about their descendants.

If we now compare the hop count for all the point-to-point connections between all the nodes, we notice the same effect for the maximum hop count as for the storing mode, because this will be a link which always has to use the route via the virtual sink. The average hop count is reduced from 8.0 to 6.26 hops. The median reduces from 8 to 6 hops, and the standard deviation decreases from 2.26 (for one sink) to 0.994 (for four sinks).

These results, however, strongly depend on the topology of the network and the placement of the initial and additional sinks. These results take into account the point-to-point connections between all the nodes of the network. If some connections between two nodes are used more intensely than other point-to-point connections, this will have a bigger impact on the network.

We also must keep in mind that RPL is optimised for multipoint-to-point connections and is less optimal for point-to-point connections.

9 Conclusion

This paper describes a method to support multiple sinks for RPL in accordance to the limited guidelines of the RPL RFC [15]. The implementation of multiple sinks using the concept of a virtual sink allows an easy deployment of additional sinks without having to change how the non-sink nodes operate. Because the adaptations are limited to the sink nodes, the compatibility with wireless sensor networks which use the existing IETF RPL protocol, is preserved. One of the pre-requirements for a reliable network with multiple sinks is the availability of a reliable network between the different sink nodes to coordinate the synchronisation between the sinks. Nevertheless, the proposed techniques are implemented in the Contiki operating system, these techniques are also applicable in other operating systems for sensor networks.

The developed support for multiple sinks is evaluated in a simulation environment and a real-life sensor network in an office environment. Results presented in this paper show that the implementation behaves as expected, i.e. the traffic load is distributed over the different sinks. The performed performance tests illustrate that the addition of extra sinks has advantages. For the simulations of this topology, the maximal energy usage of the nodes decreased with about 45% and the average energy consumption decrease with more than 30%. For the testbed experiments, the average energy consumption decreased from 2.356 to 1.558 mW (so 33.87%), and the maximal energy consumption reduced with more than 50% when increasing the number of sink nodes from 1 to 2.

In the simulations the maximal packet loss decreased from about 60% to about 26%, and the average packet loss decreased from 25.94% to 10.17% when increasing the

number of hops from one to four sinks. In the testbed experiments, the average packet loss decreased from 57.72% to 22.15%. The maximal packet loss decreased about 30% when increasing the number of sink nodes from 1 to 2.

Of course, also the positioning of the extra sinks is of importance which is illustrated by the research cited in Section 2. This paper however did not focus on the positioning of the sink nodes. In the tests on the testbed, the nodes were located on the opposite sides of the building. Compared to the situation of the only sink node, the children leaf nodes at the bottom of the tree became the direct neighbours of the second sink node.

The percentages presented depend on the topology and the positioning of the sinks nodes. Therefore, the focus is not on the presented percentages but on the trend of the effect of the optimisations. When choosing a different positioning of the sink, the results can be improved or can even result in lower gains. So depending on the characteristics of the network and the position of the sinks, high performance gains can be achieved when adding additional sinks to a wireless sensor network.

As presented, the use of multiple sinks, interconnected by a virtual sink, can lead to a shorter hop distance between the nodes of the network and the actual sink. This can also benefit the end-to-end delay for this type of communication for normal traffic loads. Like with the hop count for point-to-point connections, some links can experience higher end-to-end delay and other links can experience smaller end-to-end delays. However, depending on the initial topology and the location of the new sinks, the overall end-to-end delay can decrease. Further research can confirm this hypothesis.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

This research is funded by the FWO-Flanders through a FWO post-doctoral research grant for EDP.

Received: 18 December 2013 Accepted: 26 May 2014

Published: 5 June 2014

References

1. KN Jie Lian, GB Agnew, Data capacity improvement of wireless sensor networks using non-uniform sensor distribution. *Int. J. Distributed Sensor Netw.* **2**(2), 121–145 (2006)
2. J Li, S Ji, H Jin, Q Ren, Routing in multi-sink sensor networks based on gravitational field, in *Proceedings of International Conference on Embedded Software and Systems, ICES'08* (IEEE Computer Society, Washington, DC, USA, 29–31 July 2008), pp. 368–375
3. H Kim, Y Seok, N Choi, Y Choi, T Kwon, *Optimal multi-sink positioning and energy-efficient routing in wireless sensor networks. Lecture Notes in Computer Science (LNCS)* vol. 3391, (Springer Berlin, Heidelberg, 2005), pp. 264–274
4. El Oyman, C Ersoy, Multiple sink network design problem in large scale wireless sensor networks, in *2004 IEEE International Conference on Communications*, vol. 6, (Paris, 20–24 June 2004), pp. 3663–3667
5. J Flathagen, O Kure, PE Engelstad, Constrained-based multiple sink placement for wireless sensor networks, in *2011 IEEE 8th International*

- Conference on Mobile Adhoc and Sensor Systems (MASS)*, Valencia, 17–22 October 2011, pp. 783–788
6. A Bogdanov, E Maneva, S Riesenfeld, Power-aware base station positioning for sensor networks, in *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM*, vol. 1 (2004), Hong Kong, 7–11 March 2004
 7. JA Hartigan, *Clustering Algorithms*, 99th edn. (Wiley, New York, 1975)
 8. Z Vincze, R Vida, A Vidacs, Deploying multiple sinks in multi-hop wireless sensor networks, in *IEEE International Conference on Pervasive Services*, (Istanbul, 15–20 July 2007), pp. 55–63
 9. H Safa, M Moussa, H Artaïl, An energy efficient genetic algorithm based approach for sensor-to-sink binding in multi-sink wireless sensor networks. *Wireless Network*. **20**(2), 177–196 (2014)
 10. SR Gandham, M Dawande, R Prakash, S Venkatesan, Energy efficient schemes for wireless sensor networks with multiple mobile base stations, in *IEEE Global Telecommunications Conference, 2003. GLOBECOM '03*, (2003), vol. 1, (San Francisco, 1–5 December 2003), pp. 377–381
 11. ZM Wang, S Basagni, E Melachrinoudis, C Petrioli, Exploiting sink mobility for maximizing sensor networks lifetime, in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences, HICSS '05*, Hawaii, 3–6 January 2005, pp. 287–287
 12. I Papadimitriou, L Georgiadis, Maximum lifetime routing to mobile sink in wireless sensor networks, in *Proceedings of the 13th IEEE SoftCOM*, Split, Croatia, 15–17 September 2005
 13. LB Saad, B Tourancheau, Sinks mobility strategy in IPv6-based WSNs for network lifetime improvement, in *International Conference on New Technologies, Mobility and Security (NTMS)*, Paris, 7–10 February 2011, pp. 1–5
 14. MI Khan, WN Gansterer, G Haring, Static vs. mobile sink: the influence of basic parameters on energy efficiency in wireless sensor networks. *Comput. Comm.* **36**(9), 965–978 (2013), Reactive wireless sensor networks
 15. 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. IETF (2012), <http://www.ietf.org/rfc/rfc6550.txt>. Accessed 6 December 2013
 16. P Levis, T Clausen, J Hui, O Gnawali, J Ko, The trickle algorithm. IETF (2011), <http://www.ietf.org/rfc/rfc6206.txt>. Accessed 4 April 2014
 17. Contiki: The open source OS for the internet of things, <http://www.contiki-os.org>, Accessed 7 April 2014
 18. JL Romkey, Nonstandard for transmission of IP datagrams over serial lines: SLIP. IETF (1988), <http://www.ietf.org/rfc/rfc1055.txt>. Accessed 21 October 2013
 19. Laboratoire de Conception et d'Intégration des Systèmes: development: tools [Wismote] (2012), <http://wismote.org/doku.php?id=development:tools>. Accessed 20 November 2013
 20. R Hinden, S Deering, Internet protocol version 6 (IPv6) addressing architecture. IETF (2003), <http://www.ietf.org/rfc/rfc3513.txt>. Accessed 6 May 2014
 21. An introduction to Cooja (2014), <https://github.com/contiki-os/contiki/wiki/An-Introduction-to-Cooja>. Accessed 13 May 2014
 22. Ultra low power IEEE 802.15.4 compliant wireless sensor module (2006), <http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf>. Accessed 15 April 2014
 23. Q Wang, M Hempstead, W Yang, A realistic power consumption model for wireless sensor network devices, in *2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, SECON '06*, vol. 1, Reston, VA, 28 September 2006, pp. 286–295
 24. S Bouckaert, W Vandenberghe, B Jooris, I Moerman, P Demeester, The w-iLab.t testbed, in *Proceedings CD of the 6th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, TridentCom '10* Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (ICST), (Ghent, Belgium, 2010)
 25. CREW Project: w-iLab.t Documentation (2013), <http://www.crew-project.eu/portal/wilabdoc>. Accessed 6 December 2013

doi:10.1186/1687-1499-2014-91

Cite this article as: Carels et al.: Support of multiple sinks via a virtual root for the RPL routing protocol. *EURASIP Journal on Wireless Communications and Networking* 2014 **2014**:91.

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com