

Article

ABORt: Acknowledgement-Based Opportunistic Routing Protocol for High Data Rate Multichannel WSNs

Hamadoun Tall¹ and Gérard Chalhoub^{2,*} 

¹ University Clermont Auvergne, 49 Boulevard François Mitterrand, BP 10448, F-63000 Clermont-Ferrand, France; hamadoun.tall@uca.fr

² CNRS, UMR 6158, Campus des Cézeaux, LIMOS, F-63173 Aubière, France

* Correspondence: gerard.chalhoub@uca.fr; Tel.: +33-4-73177048

Received: 17 August 2017; Accepted: 7 October 2017; Published: 13 October 2017

Abstract: The ease of deployment and the auto-configuration capabilities of Wireless Sensor Networks (WSNs) make them very attractive in different domains like environmental, home automation or health care applications. The use of multichannel communications in WSNs helps to improve the overall performance of the network. However, in heavy traffic scenarios, routing protocols should be adapted to allow load balancing and to avoid losing data packets due to congestion and queue overflow. In this paper, we present an Acknowledgement-Based Opportunistic Routing (ABORt) protocol designed for high data rate multichannel WSNs. It is a low overhead protocol that does not rely on synchronization for control traffic exchange during the operational phase of the network. ABORt is an opportunistic protocol that relies on link layer acknowledgements to disseminate routing metrics, which helps to reduce overhead. The performance of ABORt is evaluated using the Cooja simulator and the obtained results show that ABORt has a high packet delivery ratio with reduced packet end-to-end delay compared to two single channel routing protocols and two multichannel routing protocols that use number of hops and expected transmission count as routing metrics.

Keywords: wireless sensor networks; multichannel routing; queueing delay; acknowledgement; synchronization

1. Introduction

Wireless sensor networks (WSNs) are a very attractive solution for their ease of deployment and auto-configuration. A wide range of needs can be answered using WSNs in environmental, medical, industrial or military applications [1]. This technology has attracted a lot of researchers and intensive work has been done on energy efficiency communication protocols [2,3], and has also been considered for high data rate applications [4,5]. The most successful standard to define physical and link layers for this domain is the Institute of Electrical and Electronics Engineers (IEEE) 802.15.4 [6]. It has paved the way for other Internet Engineering Task Force (IETF) standards such as the Routing Protocol for low-power and Lossy networks (RPL) [7] and industrial standards such as ZigBee [8], WirelessHART [9] and ISA100.a [10]. In this paper, we focus on enhancing the network efficiency in high data rate scenarios in terms of packet delivery ratio, throughput, and end-to-end delay by investigating the use of an adaptive routing protocol over a multichannel MAC (Medium Access Control) protocol.

The use of multiple channels helps to mitigate some issues encountered in single channel communications like interference and collisions caused by high contention. With more bandwidth, less interference and less collisions, the network performances will be enhanced and the traffic load in the network will increase. Nodes will have more data to transmit towards the sink and intermediate

nodes will have to cope with the traffic rate and find a way to balance the traffic load among them. This is typically the role of a load balancing routing protocol.

Routing in multichannel WSNs has been extensively studied [11,12]. However, having a routing protocol that permits reaching a high data rate in multichannel WSNs remains an open research topic. The constrained resources (computation, memory, energy and communication range) of sensor nodes coupled with a shared wireless medium makes the issue more complex. In single channel WSNs, the interference level is high under high data rate scenarios, which leads to higher risks of collisions and packet loss. Multichannel communication has been introduced in order to reduce the interference level and improve the data delivery ratio. In multichannel protocols, nodes do not listen on the same channel all the time like in single channel protocols. This makes topology and routing information diffusion more challenging. To overcome this issue, some protocols introduce a synchronization phase where all nodes in the network have to tune their radio on the same channel for exchanging control traffic. However, synchronization is time and energy consuming in addition to the fact that it is hard to achieve in large scale networks.

In this paper, we propose an alternative approach for routing information sharing in multichannel communication by using MAC layer acknowledgements in an opportunistic way. ABORt, or Acknowledgement-Based Opportunistic Routing, is our contribution. It is a routing protocol designed for high data rate multichannel WSNs. ABORt does not rely on periodic control messages for exchanging the routing information. Hence, it avoids costly synchronization periods during the operational phase of the network for exchanging control traffic and enhances throughput and access delay by doing so. The routing metric used by ABORt is based on the average queueing delay of data packets. In addition, ABORt uses multiple next hop nodes towards the destination whenever possible and efficient to do so in order to achieve load balancing.

The remainder of the paper is organized as follows. In Section 2, we present the main contributions in the related work on joint channel selection and routing protocols in multichannel WSNs. In Section 3, we present a detailed description of ABORt. In Section 4, we discuss the simulation results. Section 5 concludes the paper and presents our future work.

2. Related Work

In [11], the authors propose a Multi-Channel Real-Time (MCRT) joint MAC and routing protocol designed for real-time communications in multichannel WSNs. It organizes the network into different partitions based on data flows such that interference among different flows can be minimized. The routing approach is based on the packet delay between the node and its upstream neighbors combined with the node transmission power. Simulation results using the Network Simulator (NS-2) show that MCRT is able to achieve better end-to-end delays and energy consumption compared to three other protocols. However, MCRT is not tested in a congested network, in the evaluation, traffic is generated by only 10% of nodes in the network generating only one packet every 4 s.

Authors of [12,13] propose a Distributed Routing and Channel Selection scheme (DRCS) for multichannel WSNs. DRCS is a joint channel assignment and quality aware routing protocol for multichannel WSNs that aims at improving network lifetime. Channel assignment and next hop neighbor selection are based on the remaining battery level and the link quality of the next hop neighbors evaluated based on the expected transmission count (ETX). DRCS performance evaluation shows that the proposed approach reduces energy consumption by avoiding overhearing and thus improves the network lifetime. However, DRCS was evaluated under low traffic rate (0.4 packets per second) as well.

In [14], a tree-based multichannel protocol (TMCP) is proposed. TMCP divides in a static manner the whole network in several sub-trees where a common communication channel is allocated to each sub-tree. The idea followed by sub-trees creation is to put the same number of nodes in different sub-trees. The routing approach is tree-based and nodes have to choose one parent that brings less

interference among all its potential parents. The approach proposed by TMCP reduces inter-tree interference, but the intra-tree interference problem remains unsolved.

In [15], the authors propose a Hybrid Multi-Channel MAC protocol (HMC-MAC) with a tree-based routing approach. The routing approach follows a tree where each node chooses a parent among its potential next hop neighbors. Simulation results using the NS-2 simulator show that, compared to the Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) and 2-hop channel assignment, HMC-MAC improves the network throughput and reduces the number of dropped packets due to a medium access problem. HMC-MAC suffers from high end-to-end delay due to the synchronization phase and high number of packets lost due to queue overflow at nodes close to the sink.

In [16], authors propose a Multi-Channel Collection (MCC) protocol that aims at optimizing data collection in multichannel WSNs. MCC is a time-scheduled protocol with globally synchronized Time-Division Multiple Access (TDMA) scheduling. Through experiments with Tmote Sky nodes, MCC shows that the network throughput can be improved by mixing channel allocation, routing approach and time scheduling. MCC increases the overall network throughput but requires a precise time synchronization, which is hard to obtain in large scale WSNs.

The authors of [17] propose a Joint Routing and Channel Assignment (JRCA) scheme for wireless mesh networks. JRCA aims at improving the quality of the communication in multi-radio multi-gateway wireless mesh networks. The routing approach is based on the multi-hop route quality in terms of end-to-end probability of success and the delay experienced by packets. Performance evaluation using NS-2 simulator shows that JRCA improves network throughput, delivery ratio and end-to-end delay compared to a single channel protocol and some existing multichannel protocols. JRCA is proposed for wireless mesh networks, and applying it in WSNs needs adaptations and modifications due to the limited capacities of sensor nodes (such as queue length of 200 packets with a packet size of 1000 bytes, in addition to throughput and channel access differences). Moreover, JRCA suffers from overhead and this aspect was not evaluated in the paper.

In [18], the authors proposed DiffQ, which is queue differential prioritization-based and aims to control congestion in wireless multi-hop networks. DiffQ uses the queue size difference between two neighboring nodes and sends traffic to the node that minimizes this queue size difference. DiffQ supposes that data packets have many destinations and relies on a routing protocols that specifies the next-hop of each packet. DiffQ is based on the current number of packets in the queues of neighbor nodes, which does not give an idea about the queueing delay of neighbors. A node with a high queueing delay might have fewer packets in its queue simply because it was not chosen as a next-hop by other nodes. In addition, DiffQ is not considered as a routing protocol, and it uses the already existing ETX [19] routing approach for next hop selection in the presented experiment. Moreover, in the presented experiment results, many packets are loss due to queue overflow, about 2500 drops over 2 min experiment duration in an IEEE 802.11 network.

The IETF Routing Over Low power and Lossy networks (ROLL) Working Group designed a routing solution for low power and lossy networks (LLNs) including wireless sensor networks. The working group has specified the Routing Protocol for Low power and lossy networks (RPL). RPL is designed to meet the core requirements for data transmission in LLNs. It is being developed as a standard to be deployed in a number of environments: urban networks, smart grid networks, industrial networks, building and home networks. The routing scheme of RPL consists of constructing a Directed Acyclic Graph (DAG) rooted at the sink, which will minimize the cost of reaching the sink from any node in the network as per the Objective Function (OF). The OF of RPL can consider the Expected number of Transmission (ETX) or the number of hops from each node towards the final destination (HopCount) to construct the DAG. Nevertheless, RPL is not designed for high data rate networks and suffers from congestion because it lacks a load balancing mechanism.

Most of the presented joint channel allocation and routing protocols either have a synchronization phase [15,16] that increases the packet end-to-end delay or overloads the network with periodic beacon frames [13] for channel allocation and routing information sharing. Our contribution,

ABORt, presented in the next section, is a routing protocol for multichannel WSNs that avoids the synchronization period once the network is in operational mode and achieves low end-to-end delays and limited overhead.

3. Acknowledgement-Based Opportunistic Routing Protocol

We consider a large-scale multichannel WSNs for data collection applications, where all collected data are destined to one common node usually called the sink in the literature. Many-to-one data collection is very common in WSNs and usually known as convergecast in the literature. In high traffic load situations, all data will converge from leaf nodes towards the sink which causes high congestion in nodes close to the sink. The used routing policy will have an effect on the delivery ratio and end-to-end delay. The routing protocol must include a load balancing approach in order to avoid overloading some nodes while others are under-loaded.

The aim of ABORt is to avoid congestion by balancing traffic load among under-loaded next hop nodes in the network. On the MAC layer, ABORt uses a multichannel protocol with semi-dynamic channel allocation like the one presented in [15]. On the network layer, the routing metric relies on the average packet queueing delay. Moreover, ABORt uses a multi-interface sink with three radio interfaces that can receive data at the same time on different channels. The sink node may have more or less than three radio interfaces depending on the traffic load of the network. In this paper, we decided to use three radio interfaces for the sink. All other nodes in the network operate with one radio interface. The following sub-sections describe in detail how ABORt operates.

3.1. Neighbor Discovery and Network Creation

When the network is started, all nodes tune their radio interface on the same communication channel. This allows each node in the network to be able to receive and broadcast information and to discover its neighborhood. Neighbor discovery helps to build the network and allows nodes to select available channels. All nodes in the network, except the sink, have to discover their 1-hop, 2-hop and 3-hop neighbors. Neighborhood discovery is done using periodic beacon frames broadcast during the start-up phase of the network. Each node builds its 1-hop neighbor list on the reception of beacon frames from its neighbors. Each node includes the list of its 1-hop neighbors in the beacon frame. This way receiving nodes can build the list of their 2-hop neighbors using the neighbors lists of their 1-hop neighbors. The list of 2-hop neighbors is also included in the beacon frames, which will be used by receiving nodes to build their list of 3-hop neighbors. At the end of the neighborhood discovery process, each node in the network except the sink has a list of its 1-hop, 2-hop and 3-hop neighbors. In the network scenario presented in Figure 1, the 1-hop, 2-hop and 3-hop neighbor lists of node 8 are coloured in green and yellow.

In this version of ABORt, we assume that the number of nodes in the network does not increase once the network starts up. Nodes integration after the network set up phase is out of the scope of this paper.

3.2. Channel Allocation Scheme

When multiple channels are used in the same network for communication, each node has to know the reception channel of its neighbors to be able to communicate with them. This requires sharing information on channel allocation among neighboring nodes. As demonstrated in [20], channel reuse in the 3-hop neighborhood leads to collisions when MAC layer acknowledgement messages are enabled. In order to prevent these kinds of collisions, nodes avoid selecting already used channels in their 3-hop neighborhood when possible.

Nodes select channels in an order based on their *ids*. A node *id* is a 2-byte integer that is allocated to each node in a unique manner at the start-up phase. The node with the smallest *id* in its 3-hop neighborhood selects its channel first. A node does not select a channel as long as its predecessor has not announced its reception channel in the beacon. A predecessor of a node is its 1-hop, 2-hop or 3-hop

count routing metric. In addition, each node builds a top-list neighbors table that contains all of its next hop neighbors with the minimum hop count route. In order to ensure load balancing, for each data packet to transmit, a node will randomly choose one neighbor in the top-list and forward its packet to this neighbor.

For each node, each generated or forwarded packet is enqueued and will be transmitted following the First In First Out (FIFO) policy. For each node, whenever a data packet is enqueued for transmission, the time is recorded using the node local clock. When the same packet is dequeued, the time is also recorded. The difference between the dequeued and enqueued instants is the packet queuing delay. We calculate the average value of the queuing delay over the last ten packets (ten packets is a value obtained through heuristics showing that smaller values create oscillations, and bigger values prevent nodes from accurately updating its delay), and we call it node delay d . If the node has not dequeued ten packets yet, d is the average value of already dequeued packets. To make the queuing delay average more representative of the recent traffic conditions, we use a weighing factor of 2 for the 5 most recently dequeued packets. The formula of node delay calculation is presented in Equation (1):

$$\text{Node delay} = \frac{\sum_{i=1}^5 \text{queueingdelay}(i) + \sum_{i=6}^{10} 2 \times \text{queueingdelay}(i)}{15}, \quad (1)$$

where *queueing delay* (i) is the difference between dequeuing instant and queuing instant of packet (i).

Once a node has already dequeued ten packets, it uses a sliding interval where the oldest queuing delay is deleted and the new one is inserted. Doing so helps to be sure that each node always records the time of the most recent dequeued ten packets. Node delay d is used to calculate path delay D . For each node, D is the average time that packets are estimated to spend from this node to the sink. Sink 1-hop neighbors have the same value of node and path delays.

3.4. ABORT Next Hop Neighbor Selection and Queue Overflow Avoidance

ABORT uses acknowledgement (ACK) messages in an opportunistic way to disseminate the routing metric in the network. We modified the default ACK frame and added 2 bytes that contain the routing metric value. When sink 1-hop neighbors transmit a packet to the sink for the first time, they calculate D and have to transmit it to their neighbors using ACK frame after receiving a packet from these neighbors. When 2-hop neighbors of the sink receive the ACK frames sent by sink 1-hop neighbors, they have the path delay of these neighbors. They select the minimum path delay and add it to their own node delay in order to calculate their path delay. These 2-hop neighbors will also transmit their path delay in ACK frames and the process continues until path delay is calculated by leaf nodes. Once all nodes in the network get a path towards the sink, each node path delay is updated whenever a smaller path delay is received or a new node delay is computed. Note that whenever a node needs to know its neighbor path delay, it has to transmit a data packet to it and waits for ACK that contains the path delay.

Nodes that are out of range of the sink node have to forward their data to a next hop neighbor, which will also forward it towards the sink. Depending on nodes' position in the network topology and the used routing metric, some nodes may have several potential next hop neighbors to forward their data towards the sink. Using ABORT, nodes build a top-list of next hop neighbors. Nodes in the top-list will be used for sending frames towards the sink node. For each frame, a next hop is chosen randomly from the top-list. This way, nodes will achieve load balancing among their selected neighbors.

The top-list contains neighbors with the smallest routing metric within a certain threshold defined in terms of milliseconds. The choice of this threshold must be done in order to choose most convenient routes and avoid routing loops from occurring. Using a heuristic approach based on simulation results, we fixed this threshold to 2 ms. Results showed that this value avoids routing loops when transmitting data to the neighbors in the top-list. Thus, in what follows, top-lists contain the neighbor with the

smallest path delay plus all neighbors that have a path delay that does not exceed the smallest path delay by 2 ms.

For each data packet, a node will randomly choose one neighbor in the top-list and switches to its reception channel for transmission. The fact that the next hop changes from one packet to another allows load balancing on a per hop basis.

In case only one neighbor is available in the top-list, a node has to trigger the top-list update process. This update process consists of transmitting a data packet to each next hop neighbor and receiving the ACK that contains this neighbor path delay. Doing so helps to get the path delay of all potential next hop neighbors and rebuild the top-list in case changes occurred in the routing metrics of neighbors. In order to avoid doing the update process endless when no other path delays are within the threshold, the update process is only triggered after successful transmission of ten data packets to the top-list member. The process ends when all neighbors are visited once.

On Figure 2, we present the top-list concept with a 12-node network scenario. The couple d/D of each node is labelled at its left. Node 9 has four next hop neighbors: nodes 1, 7, 8 and 4. The smallest path delay of 9 is 10 ms via node 4. Its top-list contains 4, 8 and 7. Each time node 9 has to transmit a packet that it randomly chooses one neighbor from its top-list and switches to this neighbor reception channel and transmits the data packet. When the next hop neighbor receives the transmitted packet, it will transmit an ACK that contains, in addition to ACK traditional fields, the node path delay. This ACK also allows all its neighbors that are listening on its reception channel to get the ACK frame and read the updated routing metric value. In this way, the routing metric is included in the ACK message in an opportunistic way every time a node acknowledges a data frame.

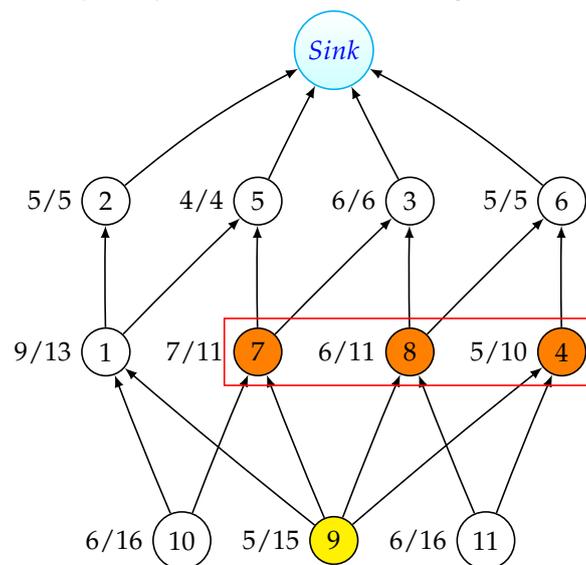


Figure 2. Node delay d and path delay D presented for each node on its left d/D . Node 9 top-list neighbors are 7, 8 and 4 encircled in red.

3.5. Queue Overflow Avoidance

Queue overflow leads to packet loss that reduces the delivery ratio. ABORT avoids queue overflow at each node. Nodes constantly monitor their queue occupancy level. When a critical occupancy threshold based on the difference between the arrival rate and departure rate of packets is reached in a queue, the node has to alert its transmitting neighbors using an alert message. When the neighbors receive the alert message, they have to *abort* using this overloaded node as a next hop and find an alternative next-hop.

4. Simulations Environment and Performance Evaluation

ABORt performance is evaluated by means of simulation using a Cooja emulator [21] with the parameters shown in Table 1. We used an area of $100 \times 100 \text{ m}^2$ where sensor nodes are randomly scattered, but we ensure that the network is connected (no isolated nodes). In Cooja, each node is emulated with the same sensor nodes capacities as sky notes. This ensures realistic parameters and behavior of nodes. Indeed, the simulation code can be used for experiments on physical sensor nodes.

In order to show that multichannel routing helps to improve the network throughput, we compared ABORt with two single channel protocols. The routing protocol for low power and lossy networks (RPL) with the ETX objective function and a Collaborative Load Balancing Algorithm to avoid queue overflow in WSNs (CoLBA) [22]. In RPL, we fixed the minimal interval for sending out Destination oriented directed acyclic graph Information Object (DIO) messages to 500 milliseconds ($I_{min} = 500 \text{ ms}$) and the number of time I_{min} is continuously doubled is 16 ($I_{doubling} = 16$). CoLBA is a queueing delay-based dynamic routing protocol that avoids queue overflow. The routing metric is exchanged using specific control frames referred to as beacons. To evaluate ABORt efficiency compared to other multichannel routing protocols, we compared it to M-HopCount and M-HopCount-Sync. M-HopCount uses a routing metric that selects the shortest path in terms of number of hops that we implemented using the same ACK-based metric exchange method of ABORt. M-HopCount-Sync also uses a routing metric based on shortest path in terms of number of hops, but node activity is organized in cycles. Each cycle is divided into two phases: a synchronization phase and a data transmission phase. During the synchronization phase, all nodes in the network tune their radio interface to the same channel to exchange beacon frames. The data transmission phase is dedicated to exchanging data packets and nodes have to switch from their reception channel to the reception channel of their neighbors. All of these protocols start with the same network set-up phase during which neighbor discovery and channel selection are done.

Note that the presented results are extracted once our proposed routing scheme has completely taken over (except for the overhead results). The duration of the transient phase from hop count routing to ABORt varies from 8 s to 1 min depending on the network size and the network depth.

Table 1. Simulation parameters.

Parameters	Values
Medium access	IEEE 802.15.4 CSMA/CA [23]
Number of channels	16
Duty-Cycle	Disabled
Radio model	MRM with random path loss
Packet queue size	8 packets
Packet size	50 bytes
Simulation surface	$100 \times 100 \text{ m}^2$
Topology	Random mesh
Number of nodes	10, 20, 40 and 80
Packet generation rate	{1, 5, 10} packets/s/node
Number of repetitions	10 iterations

4.1. Packet Delivery Ratio

We calculated the delivery ratio as the ratio between the number of received packets by the sink and the overall generated data packets in the network. Results presented on Figure 3 show that, in light data traffic scenarios, where each node generates one packet per second, the delivery ratios of all protocols are high (between 92% and 99%) and very close with a slight advantage for ABORt. Indeed, when the network is under-loaded, the medium is free most of the time, which results in collision-free transmissions. Thus, using multiple channels is not very useful and optimizing the routing protocol has very little impact. When traffic load is higher (generation of five or ten packets per second per

node), the advantage of ABORt becomes more important as shown in Figures 4 and 5. The low delivery ratio of RPL and CoLBA is mainly due to high contention, interference and collision, leading to packet loss in a single channel shared medium. The delivery ratio of CoLBA outperforms RPL and the gap increases with the packet generation rate. Indeed, CoLBA is based on a load balancing routing and queue overflow avoidance that helps to mitigate packet loss due to queue overflow in a high traffic load scenario from which RPL suffers. The delivery ratio gap between ABORt and M-HopCount varies between 6 and 20 points. The low delivery ratio of M-HopCount and M-HopCount-Sync compared to ABORt is due to their static routing metric. In a static network, the minimum number of hops from each node to the sink is static and the shortest path from each node towards the sink is always the same. Nodes always use the same next hop to forward their data packets. Thus, some nodes are overloaded, which provokes congestion, queue overflow and packet loss leading to low delivery ratio.

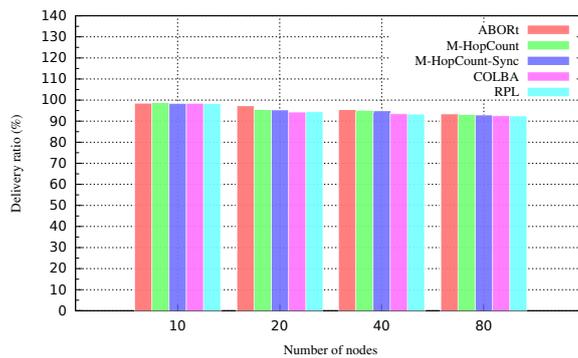


Figure 3. Packet delivery ratio for 1 pkt/sec/node.

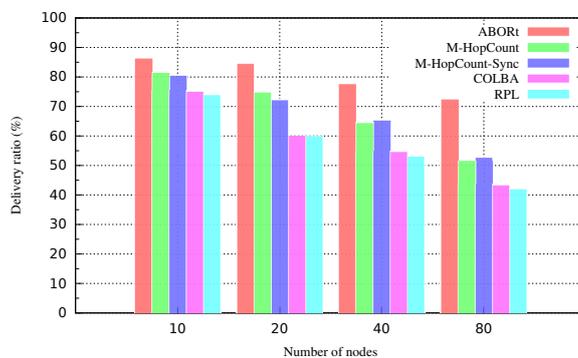


Figure 4. Packet delivery ratio for 5 pkts/s/node.

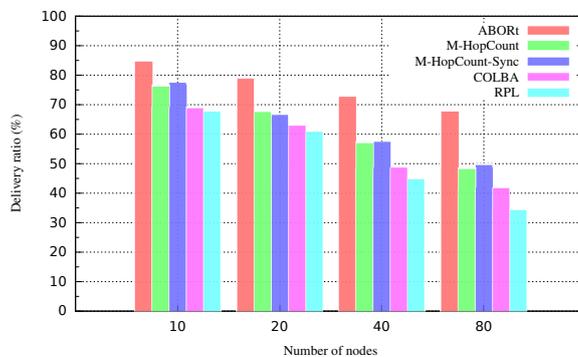


Figure 5. Packet delivery ratio for 10 pkts/s/node.

4.2. Packet Loss Due to Queue Overflow

One of the negative effects of congestion in WSNs is queue overflow, which leads to packet loss. The number of dropped packets due to queue overflow reflects the congestion level in the network. Figures 6 and 7 present the ratio of lost packets in log scale compared to the total number of generated packets in the network for 5 pkts/s/node and 10 pkts/s/node, respectively. Note that we obtained 0 lost packets for scenarios with 1 pkt/sec/node for all protocols.

Results of Figure 6 show that the number of lost packets due to queue overflow for ABORt and CoLBA is almost null. In the worst case of simulated scenarios, CoLBA loses 0.042% and ABORt 1.09% of the generated packets. CoLBA is a routing protocol in single channel network; thus, all nodes in the network communicate using the same channel. This facilitates broadcasting an alert message to the transmitting nodes when the receiving node queue is nearly full. However, ABORt uses multiple channels; thus, the broadcast support is more complex and nodes do not get the alert message at the same time. This is the main reason why CoLBA outperforms ABORt in the scenario of 80 nodes with 5 and 10 packets per second. The number of lost packets due to queue overflow of RPL, M-HopCount and M-HopCount-Sync is higher. RPL is a single channel routing protocol, so there are no parallel transmissions in the same communication range. With high traffic load, the contention level becomes higher and generated or forwarded data packets spend more time in queues, which increases the risk of overflow. Moreover, the load balancing approach of RPL is only metric based, which is not enough to fairly distribute data traffic on per hop basis. Some nodes are overloaded leading to queue overflow while others are under-loaded. M-HopCount and M-HopCount-Sync use static routing metrics. This overloads nodes that are part of the shortest paths of many nodes towards the sink. When the packet generation is high, this leads to queue overflow. Moreover, unlike ABORt and CoLBA, M-HopCount and M-HopCount-Sync do not use an alert message to inform transmitting nodes when the receiving node is suffering from queue overflow.

4.3. End-to-End Packet Delay

The end-to-end delay is the time difference between the reception instant of the packet by the sink and its generation instant by the source node. Figures 8–10 present the end-to-end delay according to the number of hops for the scenario of 40 nodes (note that all the results of the other scenarios have the same average tendency). Results show that M-HopCount-Sync end-to-end delay is higher than all other protocols because it uses a synchronization phase for beacon exchange. The synchronization lasts 2 s in order to reach all nodes of the network and during that time data packets are not sent. We also notice that except M-HopCount-Sync, the end-to-end delay of the four other protocols is close with a slight advantage for RPL and CoLBA. Indeed, RPL has a slightly lower delivery rate, which means that more packets are lost and thus packets that are not lost have less competition and waste less time in packet queues. As for CoLBA, its load balancing approach and delay-based metric help reduce the end-to-end delay.

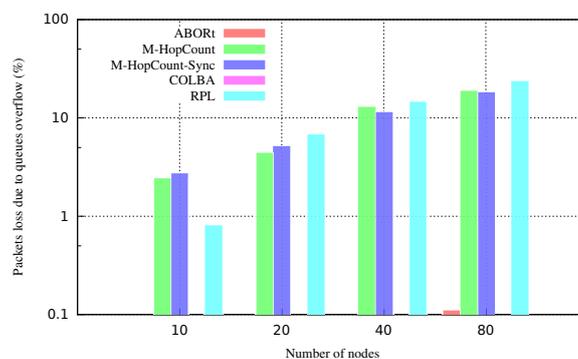


Figure 6. Queue overflow ratio for 5 pkts/s/node.

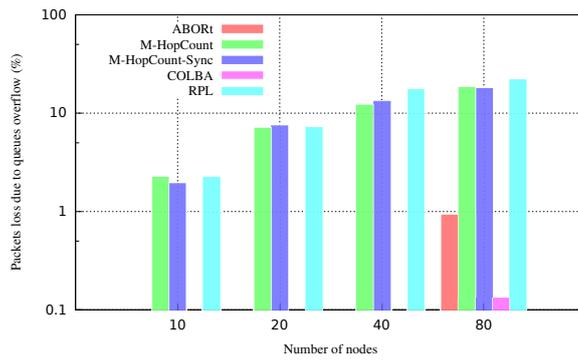


Figure 7. Queue overflow ratio for 10 pkts/s/node.

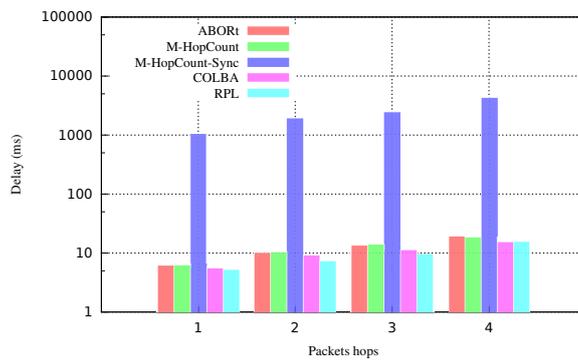


Figure 8. End-to-end delay per hop for 1 pkt/sec/node.

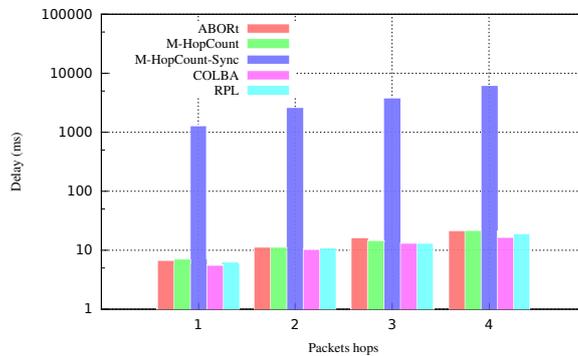


Figure 9. End-to-end delay per hop for 5 pkts/s/node.

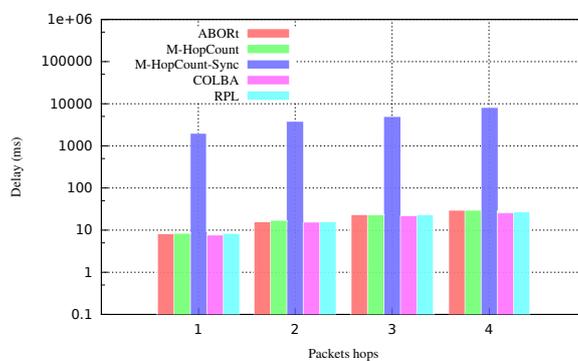


Figure 10. End-to-end delay per hop for 10 pkts/s/node.

4.4. Overhead

At the network set-up phase, nodes need to exchange control messages (beacon frames) to discover their neighborhood, in order to build the network and also to allocate channels in the cases of ABORt, M-HopCount and M-HopCount-Sync. The use of control messages leads to a certain amount of overhead, which has a direct impact on channel congestion and energy consumption. Figures 11–13 present the number of generated beacon frames for each protocol according to the number of nodes and packet generation rate. Notice that CoLBA overhead increases with the number of nodes and packet generation rate. This is because, when the traffic load becomes high, more packets are enqueued and CoLBA metric value varies more often. Thus, each node needs to transmit additional beacon frames to inform its neighbors about its new metric value or to alert them about the risk of its queue overflow. We also notice that, in the scenarios with less nodes and light traffic, CoLBA has less overhead than the other protocols because the medium is less occupied and CoLBA metric value is almost stable with a low risk of queue overflow.

Results also show that the overhead of RPL, ABORt and M-HopCount are almost the same for each network size regardless of the packet generation rate. In RPL, the overhead is mainly due to DIO, Destination Advertisement Object (DOA), DAO-ACK and Destination oriented directed acyclic graph Information Solicitation (DIS) messages. The number of DIO messages is largely higher than the others because DIO messages are used to build and maintain the network topology. DIO messages generation is governed by the trickle algorithm [24] timer regardless of the data traffic generation rate. This is why the overhead of RPL is not much affected by the packet generation rate. Thus, the overhead of RPL remains stable for the same number of nodes. In M-HopCount, beacon frames are generated only at the network start-up phase. However, in ABORt, in addition to the set-up phase, beacon frames can be generated during the data transmission in case of a high risk of queue overflow. In that case, beacon frames are sent to alert neighbor nodes to stop transmitting packets to overloaded neighbors. In all three packet generation rates, most of the time, ABORt generates more beacon frames than M-HopCount. In some scenarios, M-HopCount has slightly more overhead than ABORt essentially due to the random behavior of beacon frames transmissions during the start-up phase using CSMA/CA. In M-HopCount-Sync, beacon frames are generated at the network start-up phase and also during each synchronization phase used for routing metric exchange. This is why M-HopCount-Sync has more overhead than ABORt and M-HopCount where the synchronization phase is not needed because the routing metric information is disseminated using ACK frames. In conclusion, the approach used by ABORt and M-HopCount, which consists of using ACK frames for routing metric dissemination helps to reduce the network overhead.

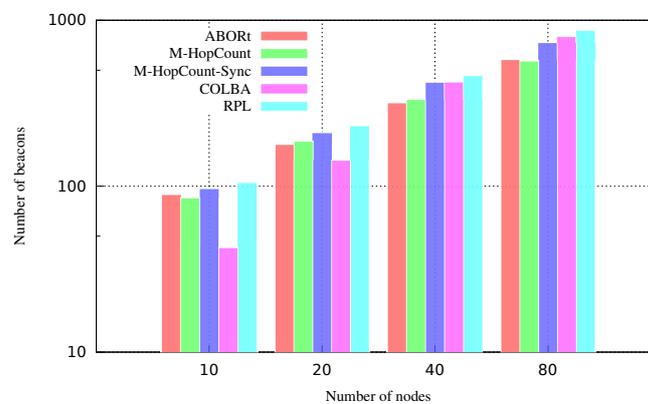


Figure 11. Overhead for 1 pkt/sec/node.

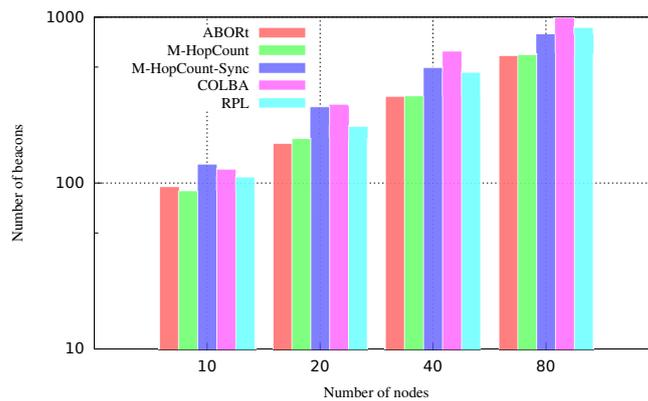


Figure 12. Overhead for 5 pkts/s/node.

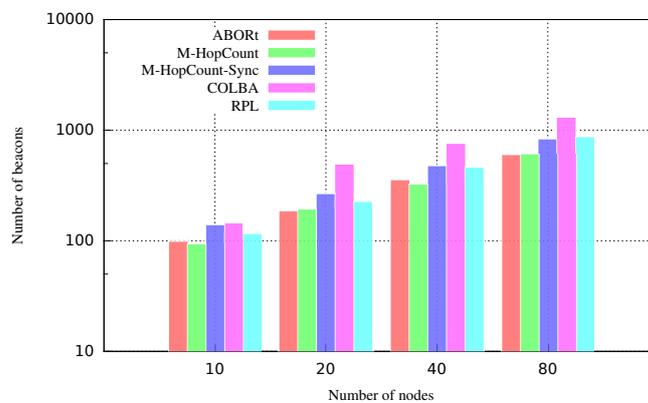


Figure 13. Overhead for 10 pkts/s/node.

5. Conclusions

The resource constrained nodes of WSNs make high data rate applications a challenging issue. In this paper, we proposed ABORt, a multichannel load balancing routing protocol that uses ACK-based control information dissemination. ABORt does not rely on a synchronization phase for network and routing information exchange. Hence, it avoids time and energy wastage in order to construct routes. It is based on a load balancing technique that helps avoid congestion in the network and a queue overflow avoidance mechanism that prevents packets from being dropped due to accumulation in packet queues. Performance evaluation using a Cooja simulator shows that ABORt is efficient in terms of delivery ratio, queue overflow, end-to-end delay and overhead compared to two single channel routing protocols

As energy is a scarce resource in WSNs, we plan on adding a duty-cycle approach allowing nodes more time to sleep with ABORt compared to other protocols. Indeed, as results showed that ABORt achieves higher delivery ratios and lower delays, we expect an energy efficient performance when used on top of a duty cycle mechanism. We also plan on enhancing ABORt in order to support node integration during the network operational phase. Adding a node to an already operational multichannel network requires coordination for knowing which channel to use in order for newcomer nodes to communicate with the network.

Acknowledgments: This research was conducted with the support of the European Regional Development Fund (FEDER) program of 2014–2020, the region council of Auvergne, and the Digital Trust Chair of the University of Auvergne.

Author Contributions: Hamadoun Tall and Gerard Chalhoub conceived and designed ABORt. They also conceived and designed the simulation scenarios and analyzed the results. They both wrote the paper. Hamadoun Tall implemented and executed the simulations.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Taruna, S.; Jain, K.; Purohit, G.N. Application Domain of Wireless Sensor Network: A Paradigm in Developed and Developing Countries. *IJCSI* **2011**, *8*, 611–617.
2. Soua, R.; Minet, P. A survey on energy efficient techniques in wireless sensor networks. In Proceedings of the IEEE joint IFIP 4th Wireless and Mobile Networking Conference (WMNC), Toulouse, France, 26–28 October 2011; pp. 1–9.
3. Rault, T.; Bouabdallah, A.; Challal, Y. Energy efficiency in wireless sensor networks: A top-down survey. *Comput. Netw.* **2014**, *67*, 104–122.
4. Zhu, N.; Mieleve, F.; Navarro, D.; Du, W.; O’connor, I. Research on high data rate wireless sensor networks. *14ème Journ. Natl. Rés. Dr. Micro Nanoélectron.* **2011**, 61.
5. Henaut, J.; Lecointre, A.; Dragomirescu, D.; Plana, R. Radio interface for high data rate wireless sensor networks. *arXiv* **2010**, arXiv:1004.0204.
6. LAN/MAN Standards Committee. *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*; IEEE Computer Society: New York, NY, USA, 2003.
7. Winter, T.; Thubert, P.; Brandt, A.; Clausen, T.; Hui, J.; Kelsey, R.; Levis, P.; Pister, K.; Struik, R.; Vasseur, J.P. *RPL: IPv6 Routing Protocol for Low Power and Lossy Networks*; IETF Standard: Fremont, CA, USA, 2012.
8. ZIGBEE, Alliance. Zigbee Specification. ZigBee Document 053474r13, 2006. Available online: http://www.olmicrowaves.com/menucontents/designsupport/zigbee/1171625602_ZigBee-Specification-2006-r13.pdf (accessed on 11 October 2017)
9. Foundation, H.C. *HART Field Communication Protocol Specification*; FieldComm Group: Austin, TX, USA, 2006.
10. International Society of Automation. *100.11 a-2009: Wireless Systems for Industrial Automation: Process Control and Related Applications*; International Society of Automation: Research Triangle Park, NC, USA, 2009.
11. Wang, X.; Wang, X.; Fu, X.; Xing, G.; Jha, N. Flow-based real-time communication in multichannel wireless sensor networks. *Wirel. Sens. Netw.* **2009**, *9*, 33–52.
12. Pal, A.; Nasipuri, A. DRCS: A Distributed routing and channel selection scheme for multi-channel wireless sensor networks. In Proceedings of the 2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), San Diego, CA, USA, 18–22 March 2013; pp. 602–608.
13. Pal, A.; Nasipuri, A. A distributed channel selection scheme for multi-channel wireless sensor networks. In Proceedings of the Thirteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing, Hilton Head, SC, USA, 11–14 June 2012; pp. 263–264.
14. Wu, Y.; Stankovic, J.A.; He, T.; Lin, S. Realistic and efficient multi-channel communications in wireless sensor networks. In Proceedings of the INFOCOM, Phoenix, AZ, USA, 13–18 April 2008; pp. 1193–1201.
15. Diab, R.; Chalhoub, G.; Misson, M. Enhanced Multi-Channel MAC Protocol for Multi-Hop Wireless Sensor Networks. In Proceedings of the IEEE Wireless Days, Rio de Janeiro, Brazil, 12–14 November 2014; pp. 1–6.
16. Chen, Y.; Gomes, P.H.; Krishnamachari, B. Multi-channel data collection for throughput maximization in wireless sensor networks. In Proceedings of the IEEE MASS, Philadelphia, PA, USA, 28–30 October 2014; pp. 443–451.
17. Pal, A.; Nasipuri, A. JRCA: A joint routing and channel assignment scheme for wireless mesh networks. In Proceedings of the IEEE IPCCC, Orlando, FL, USA, 17–19 November 2011; pp. 1–8.
18. Warriar, A.; Ha, S.; Wason, P.; Rhee, I.; Kim, J.H. DiffQ: Differential backlog congestion control for wireless multi-hop networks. In Proceedings of the IEEE SECON, San Francisco, CA, USA, 16–20 June 2008; pp. 585–587.
19. Couto, D.S.J.D.; Aguayo, D.; Bicket, J.; Morris, R. A high-throughput path metric for multi-hop wireless routing. *Wirel. Netw.* **2005**, *11*, 419–434.
20. Diab, R.; Chalhoub, G.; Misson, M. Hybrid multi-channel mac protocol for wireless sensor networks: Interference rate evaluation. In Proceedings of the IEEE VTC Fall, Las Vegas, NV, USA, 2–5 September 2013; pp. 1–6.
21. Osterlind, F.; Dunkels, A.; Eriksson, J.; Finne, N.; Voigt, T. Cross-level sensor network simulation with COOJA. In Proceedings of the IEEE LCN, Tampa, FL, USA, 14–16 November 2006; pp. 641–648.

22. Tall, H.; Chalhoub, B.; Misson, M. CoLBA: A Collaborative Load Balancing Algorithm to avoid queue overflow in WSNs. In Proceedings of the IEEE DSDIS, Sydney, NSW, Australia, 11–13 December 2015; pp. 682–687.
23. Tall, H.; Chalhoub, G.; Misson, M. Implementation and performance evaluation of IEEE 802.15.4 unslotted CSMA/CA protocol on Contiki OS. *Ann. Telecommun.* **2016**, *71*, 517–526.
24. Levis, P.; Patel, N.; Culler, D.; Shenker, S. Trickle: A self-regulating algorithm for code maintenance and propagation in wireless sensor networks. In Proceedings of the USENIX NSDI, San Francisco, CA, USA, 29–31 March 2004; pp. 15–28.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).