Evaluation of Available Bandwidth as a Routing Metric for Delay-Sensitive IEEE 802.15.4-based Ad-Hoc Networks

Muhammad Omer Farooq*, Thomas Kunz†, Cormac J. Sreenan*, and Kenneth N. Brown*
*CTVR, Department of Computer Science, University College Cork, Ireland
Email: omer.farooq@insight-centre.org, cjs@cs.ucc.ie, k.brown@cs.ucc.ie
†Department of Systems and Computer Engineering, Carleton University, Canada
Email: tkunz@sce.carleton.ca

Abstract

In this paper, we evaluate available bandwidth as a routing metric for IEEE 802.15.4-based ad-hoc networks. The available bandwidth on a data forwarding path is an approximation of the forwarding path's residual data relaying capacity. High available bandwidth on a data forwarding path implies low data traffic load on the path, therefore data flows may experience low delay and high packet delivery ratio (PDR). Our aim is to evaluate available bandwidth as a routing metric. We present different available-bandwidth-based routing protocols for IEEE 802.15.40-based networks, namely: end-to-end available-bandwidth-based routing protocol (ABR), available bandwidth and contention-aware routing protocol (ABCR), and shortest hop-count and available-bandwidth-based opportunistic routing protocol (ABOR). Moreover, we also present variants of ABR and ABCR capable of distributing a flow's data packets on multiple paths by maintaining the top K downstream nodes (the downstream nodes that advertised best data forwarding paths towards a sink node) corresponding to each sink node in a routing table. We focus on both single-sink and multi-sink networks. We performed extensive simulations, and the simulation results demonstrate that the available bandwidth routing metric shows better results when combined with a routing metric that helps to limit a data forwarding path's length, i.e., shortest hop-count or intra-flow contention count. For multi-path data forwarding towards the same sink node, and at high traffic volumes, the available bandwidth metric demonstrates best performance when combined with the shortest hop-count routing metric.

Keywords: IEEE 802.15.4; Available-bandwidth-based routing; Ad-hoc networks; Un-slotted CSMA-CA

1 Introduction

The IEEE 802.15.4 standard is being actively used in wireless networks, e.g., industrial control applications [1]. In this research, our vision of IEEE 802.15.4-based ad-hoc network is depicted in Figure 1. The example network shown is used to monitor traffic on a network of roads. Such a network can generate delay-sensitive data flows, e.g., vehicle tracking and traffic load on different roads. Afterwards, the data is forwarded to one of the sink nodes, and if required the sink node forwards data to the network controller, which we assume has a high speed connection to the sink.

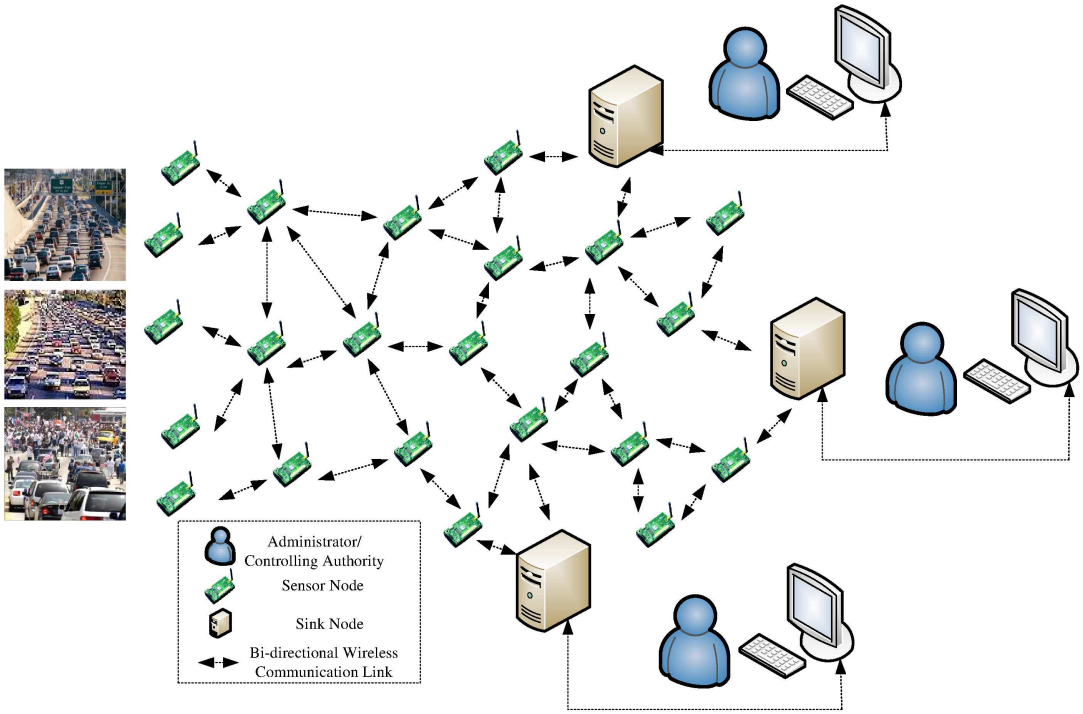


Figure 1: An Example of an IEEE 802.15.4-based Ad-hoc Network

A routing protocol discovers a data forwarding path from a source node to a destination node, and the state of links on the discovered path impacts the delay and packet delivery ratio (PDR) requirements of a data flow. E.g., if a routing protocol discovers a path with congested links, it is highly likely that a flow may experience high end-to-end delay and low PDR. In order to discover a path, the routing protocol uses a routing metric, e.g., the shortest hop-count routing metric. The choice of a routing metric can impact the quality of the discovered paths. For example, paths generated based on the shortest hop-count metric may include links with excessive data traffic, and that the flow may experience high delay and low PDR. The number of paths that a routing protocol discovers and maintains towards a destination node can also impact the performance of a flow. A routing protocol that discovers and maintains multiple best paths, and afterwards distributes a flow's data packets on those multiple paths may demonstrate better performance compared to a routing protocol that forwards the flow's data packets on a single best path.

The bandwidth supported by a communication standard determines the standard's data transmission rate. In IEEE 802.15.4-based networks bandwidth is a shared resource, as nodes within each other's interference range share the bandwidth. Therefore, the data generation rate of a node impacts the amount of bandwidth available to other nodes within the node's interference range. In ad-hoc wireless networks, a carrier sense multiple access collision avoidance (CSMA-CA) MAC layer protocol consumes bandwidth, e.g., a node can not transmit while it is waiting in a back-off mode due to busy channel and packet losses [2, 3]. We focus on the available bandwidth as a routing metric because it is an approximation of the residual data relaying capacity of a wireless channel, and it implicitly takes into account the wireless channel condition. E.g., as the number of packet losses increases the back-off duration also increases, hence the available bandwidth decreases [2, 3]. Moreover, it can capture dynamic changes in wireless

channel conditions and data traffic load inside a network. Therefore, the available-bandwidth-based routing metric can help a routing protocol to discover data forwarding paths that may help to improve the performance of data flows.

In this paper, we present an evaluation of available bandwidth as a routing metric in single-sink and multi-sink IEEE 802.15.4-based ad-hoc networks. The following are our main contributions:

1. Different ways of utilization the available bandwidth as a routing metric, e.g., considering the end-to-end available bandwidth, combining the available bandwidth with other routing metrics, i.e., shortest hop-count and intra-flow contention count.

2. Using the presented routing metrics for discovering and maintaining single and multiple data forwarding paths in single-sink and multi-sink networks.

3. Discovering the limits of the available bandwidth as a routing metric, i.e., an available-bandwidth-based routing metric demonstrates better results when combined with other routing metrics, e.g., shortest hop-count and intra-flow contention count. For multi-path data forwarding, and at high traffic volumes an available-bandwidth-based routing metric demonstrates best results when combined with shortest hop-count routing metric.

The remainder of this paper is organized as follows. Related work is discussed in Section 2. Our available-bandwidth-based routing protocols for WSNs are presented in Section 3. Simulation results are discussed in Section 4, and conclusions are presented in Section 5.

2 Related Work

In this section, we survey state-of-the-art routing protocols for ad-hoc wireless networks. Generally, the routing protocols for IEEE 802.15-4.-based ad-hoc networks can be categorized as reactive or proactive protocols. Most of the proactive protocols can be categorized as follows: opportunistic, heuristics-based, and gradient-field-based. Figure 2 shows the different categories of routing protocols for IEEE 802.15.4-based networks along with some existing routing protocols in each category.

Reactive routing protocols discover data forwarding paths towards a sink node on-demand using controlled flooding of a route request message. Mostly, reactive protocols use one or combination of the following as routing metrics: delay, reliability, hop-count, congestion, and energy. Generally, the sink selects a forwarding path. The routing protocols presented in [4, 5] are examples of reactive routing protocols. Reliable fault-tolerant multi-path routing protocol [4] (RFTM) aims at improving reliability of data packets. In a route request message, a source node broadcasts an application's reliability requirement. Considering the application's reliability requirements and the condition of a network, the sink selects multiple disjoint paths between the source and sink nodes. Based on erasure coding a data packet is partitioned, and different partitions of the packet are forwarded on the multiple disjoint paths. Multi-objective reliable multi-path routing (MRFTM) [5] is an extension of RFTM, which also considers delay and energy routing metrics along with the reliability metric.

In proactive routing protocols, each node periodically broadcasts a route advertisement message. In the message, each node advertises the sink discovered by the node along with the corresponding cost. A node can readily forward data packets to the sink using a proactive routing protocol. But, the feature comes at an expense of periodic broadcast of the route advertisement message.

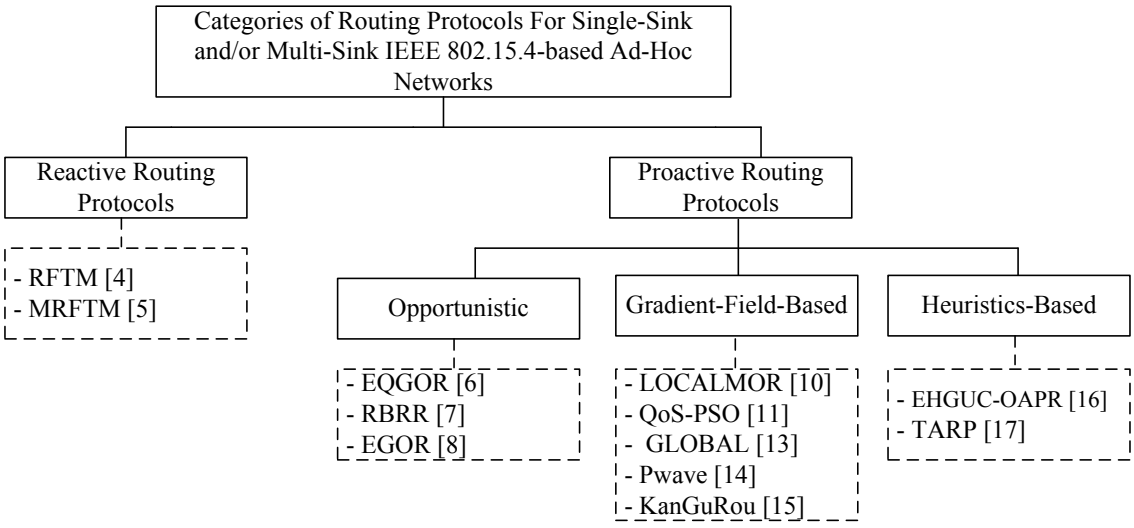


Figure 2: Categories of Routing Protocols for IEEE 802.15.4-based Ad-hoc Networks

The routing protocols presented in [6–9] are opportunistic routing protocols designed for IEEE 802.15.4-based ad-hoc networks. The Opportunistic routing protocols work as follows. A node broadcasts a data packet, and the candidate downstream node with highest priority, if it successfully receives the packet, rebroadcasts the packet. All other candidate downstream nodes, upon hearing the broadcast of the same packet, drop the packet. If a candidate downstream node with a lower priority does not hear the same packet within a certain time duration, the candidate downstream node rebroadcasts the packet. The priority of a node is derived from a routing metric, and examples of the most commonly used metrics are delay, energy, and distance. The shortcoming of the opportunistic routing protocols is that the protocols make local optimal decisions. The local optimal decisions may not be globally optimal.

The routing protocols presented in [10–15] are gradient-based routing protocols for IEEE 802.15.4-based ad-hoc networks. The gradient is the cost of using a node as a relay. The gradient field to the sink is constructed using a combination of the following routing metrics: shortest hop-count, residual energy, buffer occupancy, and delay. A gradient-based routing protocol either constructs the gradient-field locally or on an end-to-end basis. If the gradient-field is constructed locally, the protocols use the metric information of a candidate downstream node. Otherwise, the protocol uses the cumulative information of all the nodes on different possible data forwarding paths towards a sink node. Therefore, if the gradient-field is constructed locally, the data flows towards the one hop neighbour node with the minimum gradient value. Otherwise, the data flows on the path with the lowest end-to-end gradient value. As a network condition changes, the gradient value at different nodes may also change. If a gradient field is constructed locally, the resultant path may not be optimal end-to-end.

The routing protocols presented in [16, 17] are heuristics-based routing protocols for IEEE 802.15.4-based ad-hoc networks. The protocols can maintain multiple data forwarding paths towards a sink node. The protocols use route advertisement message to construct a path to the sink. Each node either advertises its own information or the node's own information and its one hop neighbours information in the route advertisement message. The protocols use optimization technique, i.e., genetic algorithm to distribute the same flow's data packets on different paths to the sink to decrease the end-to-end delay or increase the end-to-end reliability. Heuristics-based

routing protocols may not always result in an optimal path selection.

There also exist available-bandwidth-based routing protocols for IEEE 802.11-based ad-hoc networks. In [18], a reactive available-bandwidth-based multipath routing protocol and admission control algorithm are presented. Using the available bandwidth metric, the protocol discovers multiple node-disjoint paths. Different portions of a flow's data traffic is admitted on different paths based on the available bandwidth at the paths. The destination node selects multiple paths, and information about complete multiple paths are transmitted in a route reply message. The IEEE 802.15.4's frame size is limited, therefore storing complete multiple paths in the route reply message is not always feasible. In [19], an available-bandwidth-based opportunistic routing protocol and admission control algorithm for IEEE 802.11-based ad-hoc networks are presented. Data packets are relayed based on the candidate downstream nodes' available bandwidth and expected transmission cost (derived from a link's success probability). The impact of the path length on the available bandwidth is not considered. In [20], an available bandwidth and interference-aware routing protocol is presented for IEEE 802.11-based ad-hoc networks. Each node discovers the best available path to the destination node, and the data packets are relayed on the single best path. In [21], the multi-radio multi-channel variant of [20] is presented. The available-bandwidth-based protocols for IEEE 802.11-based networks can not be used in IEEE 802.15.4-based networks because they use different channel rates and slot times, and channel rate and slot time affect the available bandwidth estimate. There does not exist any research for IEEE 802.11-based networks that evaluates the available bandwidth as a metric by doing the following: combining the metric with other widely used metrics, e.g., hop-count, and using single best path and multipath data forwarding. Results for IEEE 802.11-based routing protocols' cannot be applied to IEEE 802.15.4-based networks because both standards use different frame size, channel rate, and transmission and carrier sensing range. Moreover, in IEEE 802.11-based ad-hoc networks all nodes can act as sources and sinks, but in IEEE 802.15.4-based ad-hoc networks there are a few sink nodes. Therefore, in IEEE 802.15.4-based networks data packets converge at a few locations, but this may not happen in IEEE 802.11-based networks. This difference can impact the performance of a routing protocol.

State-of-the-art research work on routing in IEEE 802.15.4-based networks focuses on designing routing protocols based on the following routing metrics: delay, reliability, congestion, hop-count, geographical distance, energy, and power consumption. The resultant routing protocols target single-sink and/or multi-sink WSNs. Moreover, the routing protocols either use single-path or multi-path data forwarding. There is no research work that focuses on extensively evaluating the available bandwidth as a routing metric in IEEE 802.15.4-based ad-hoc networks. Therefore, we evaluate the available bandwidth as a routing metric by combining it with other widely used routing metrics. Afterwards, we use the resultant routing metrics for single-path and multi-path data forwarding in single-sink and multi-sink IEEE 802.15.4-based ad-hoc networks.

3 Available-Bandwidth-based Routing Protocols

In this section, we present different ways in which we use the available bandwidth as a routing metric. We also present a distinct available-bandwidth-based routing protocol corresponding to each way the available bandwidth is used as a routing metric. For available bandwidth estimation, we use the available bandwidth estimation algorithm presented in [22]. For the readers convenience we briefly summarize the same available bandwidth estimation algorithm in the following sub-section.

3.1 Available Bandwidth Estimation Algorithm for IEEE 802.15.4-based Ad-hoc Networks

The available bandwidth estimation algorithm assumes that a network is well-connected, and nodes within two hops distance constitute the interference range of each node. Therefore, each node maintains a list of nodes within its interference range, moreover as per the algorithm, each node keeps track of the data generation rate (which also includes the amount of data that the node is forwarding for other nodes' flows apart from the node's own data) of nodes within its interference range. HELLO messages are used to discover one-hop neighbours, each node advertises its data generation rate in the HELLO message. To facilitate the discovery of two-hop neighbours at each node, every node broadcasts its one-hop neighbours addresses along with their data generation rates in a message called one-hop neighbours information message. The HELLO and one-hop neighbours information messages are also used by the nodes to advertise their own and their neighbours' available bandwidth. The summation of a node's own data generation rate and the data generation rates of the nodes within its interference range is the total data generation rate within the interference range of the node.

Apart from the data generation rates of nodes, the IEEE 802.15.4's un-slotted CSMA-CA MAC layer protocol also impacts the available bandwidth (because nodes are not synchronized in an ad-hoc network). E.g., a node is not allowed to transmit in a back-off mode, similarly a node is not allowed to transmit for the time duration the node is waiting for an ACK (acknowledgement) of a transmitted packet. Moreover, an ACK frame and retransmissions consume the bandwidth. Therefore, the available bandwidth estimation algorithm keeps track of the bandwidth consumed by the MAC layer operation per unit time (every second). Apart from the retransmitted frames and ACK frames other MAC layer overheads (back-off and ACK waiting time duration) are measured in time. The available bandwidth estimation algorithm multiplies the overhead measured in time with the channel rate to convert the overhead to bits per second (bps). Finally, the available bandwidth estimation algorithm uses Equation 1 to estimate the available bandwidth per unit time.

$$\omega_n = \rho - \left(\frac{\sum_{\mu=1}^{\theta} \beta_\mu + \gamma_\mu}{\theta} \right) bps \tag{1}$$

To estimate the available bandwidth, the available bandwidth estimation algorithm uses the sliding-window-based averaging mechanism to cope with the wireless channel impairments (reflection, refraction, diffraction, and multi-path fading), due to these impairments nodes' data generation rates and MAC layer overhead fluctuate per unit time. In Equation 1, ω_n represents the average available bandwidth in bps at any node n, θ represents the current size of the averaging window (the maximum value of θ is α, and through experiments it is shown in [3] that 5 is a suitable value for α), β_μ represents the total data generation rate within the interference range of the node at the μ^{th} index of the averaging window, γ_μ represents the total MAC layer overhead at the μ^{th} index of the averaging window, and ρ represents the channel rate.

3.2 End-to-End Available-Bandwidth-Based Routing Protocol

In our design of the routing protocol, we assume that a set N denotes all the nodes inside a network. A single node in set N is represented as n. The set of nodes within the interference range of any node n is denoted by I_n, and it is defined as: $I_n = \{m : hopcount_{n,m} \leq 2\}$.$hopcount_{n,m}$ represents the number of hops between node n and node m. B_n is the net available bandwidth at node n, and $B_n = \{min(\omega_n \forall n \in I_n)\}$. Different sink nodes inside a network are denoted by a set C, and $(C \subset N)$. A single sink node in set C is denoted by letter c. At any node

n inside a network, the available data forwarding path(s) towards a particular sink node c is denoted by $(K_c)_n$. In our routing protocol design, we consider two paths are distinct, if the sequence of nodes in the paths are not identical. A single path in $(K_c)_n$ is denoted by $(k_{c,j})_n$. A relaying node (including source node) on a path $(k_{c,j})_n$ is denoted as r. The end-to-end available bandwidth on any path $(k_{c,j})_n$ towards a sink node c is $B_{(k_{c,j})_n}$ and it is defined as $B_{(k_{c,j})_n} = \{min(B_r \forall r \in (k_{c,j})_n)\}$.

The aim of the end-to-end available-bandwidth-based routing protocol is for each node to discover a data forwarding path (if present) towards each sink node in a network such that the selected path has the highest end-to-end available bandwidth, and there are no routing loops.

3.2.1 Data Forwarding Path Discovery and Maintenance

Every node maintains a table of information about the sinks the node has discovered during the process. The table is called sink table, and it is denoted by C_{Table}. Each row in C_{Table} includes a sink's address, its sequence number, the next hop on the path to the sink, and the maximum available bandwidth on the path.

Each sink node in set C broadcasts the sink sequence number in the HELLO message apart from the information required for estimating the available bandwidth as discussed in Section 3.1. A sink node increments the sequence number each time it broadcasts the HELLO message. On the reception of the HELLO message, any direct neighbour n of the sink node extracts the information about the sink node, from the HELLO message. It matches the extracted sink node's address with the sink nodes' addresses present in C_{Table}. If the extracted sink node address does not match, the node inserts a new record in C_{Table}, for the newly discovered sink node. Otherwise, the node compares the sequence number extracted from the HELLO message with the sequence number stored in the sink table record corresponding to the sink node. If the received sink sequence number is greater, the record is updated with the received information, otherwise the sink information present in the HELLO message is ignored.

Periodically, each node in set N broadcasts a HELLO message (the same HELLO message as discussed in Section 3.1). The HELLO message broadcasted by any node n in set N contains information about those sinks in set C that node n has discovered, along with the information required by the available bandwidth estimation algorithm. The HELLO message contains the following information about the sink nodes that the node n has discovered: end-to-end minimum available bandwidth ($B_{(k_{c,j})_n}$), and the sink sequence number.

When a node n receives the HELLO message from a non-sink node m, node n extracts each sink node's information from the HELLO message. Corresponding to each sink node's information, node n matches the extracted sink node address with the addresses present in the records of C_{Table}. If the address does not match, node n inserts a new record in C_{Table} and stores the newly discovered sink node's address, sequence number, maximum available bandwidth on the path to the sink node ($B_{(k_{c,j})_m}$), and the address of next hop on the path to the sink (node m address) in it. If the extracted sink node address matches with any of the sink node addresses in C_{Table}, node n compares the extracted sink sequence number with the sink sequence number in the corresponding record of C_{Table}. If the extracted sink sequence number is less than or equal to the stored sink sequence number, the sink information is ignored. Otherwise, node n updates the sink sequence number in the corresponding record of C_{Table}. Afterwards, node n matches node $m's$ address with the next hop address stored in the corresponding record of C_{Table}. If both addresses match, node n updates the corresponding record in C_{Table} with the received sink information including $B_{(k_{c,j})_m}$. If the addresses do not match, node n compares the $\min(B_n, B_{(k_{c,j})_m})$ with the stored $B_{(k_{c,j})}$ in the corresponding record of C_{Table}, and if $\min(B_n, B_{(k_{c,j})_m})$ is greater than the stored $B_{(k_{c,j})}$, node n updates the maximum available bandwidth ($B_{(k_{c,j})}$) and next

Algorithm 1: ABR HELLO Message Processing

```
1  sink_record = Null;
2  for ∀rec_i ∈ hello_msg do
3      sink_record = is_sink_in_C_Table(rec_i.addr);
4      if sink_record == Null then
5          add_new_record_in_C_Table(rec_i);
6      end
7      else
8          if sink_record.seq_no ≥ rec_i.seq_no then
9              continue;
10         end
11         else
12             update_seq_no_in_C_Table(sink_record, rec_i.seq_no);
13             if sink_record.downstream_addr == hello_msg.src_addr then
14                 update_record_in_C_Table(sink_record, rec_i);
15             end
16             else
17                 if min(B_n, rec_i.B_(k_c,j)_m) > sink_record.B_(k_c,j) then
18                     sink_record.downstream_addr = hello_msg.src_addr;
19                     sink_record.B_(k_c,j) = min(B_n, rec_i.B_(k_c,j)_m);
20                 end
21             end
22         end
23     end
24 end
```

hop fields in the corresponding record of C_{Table}. A record from C_{Table} is deleted in any of the following cases: (i) next hop for the sink node is removed from the direct neighbour table (the direct neighbour table was discussed in Section 3.1) and (ii) if the node does not receive the discovered sink information from the node whose address is stored in the next hop address field of C_{Table} within a pre-defined time interval. Algorithm 1 summarizes the processing of the HELLO message, and in the algorithm rec_i represents a record corresponding to a sink node in the HELLO message.

Once a flow or set of flows originating from node n start transmitting data packets towards a sink node c, the end-to-end available bandwidth on the data forwarding path $(k_{c,j})_n$ decreases. Therefore, it is possible that the routing protocol may discover an alternative path $(k_{c,l})_n$ towards the same sink node such that $B_{(k_{c,l})_n} > B_{(k_{c,j})_n}$. In such an event, the routing protocol updates C_{Table} so that a node starts to use the newly discovered path that offers higher end-to-end bandwidth. But, in this case, all the data for sink c originating from node n will be using the new discovered path, i.e., $(k_{c,l})_n$. The following deficiencies are associated with this technique: (i) re-routing all data traffic on $(k_{c,l})_n$ can cause congestion on $(k_{c,l})_n$, hence data flows may experience poor performance and (ii) inefficient utilization of different paths (if present). Our solution to these deficiencies is that, once a flow starts to use a path, it is not allowed to change the path. The flow can only change the path in case of a route failure. With this there arises a need to distinguish between the paths in use by different flows originating from the same source node towards the same sink node. Our solution to this problem is to use a separate forwarding table for established flows. We assume that each flow has a unique identifier, and the identifier

will not be re-used immediately. A forwarding table at any node n in set N is denoted by $T_{Forward}$. A record in $T_{Forward}$ consist of the following: source node address, sink node address, unique flow identifier, downstream node address, upstream node address. When the forwarding module of the routing protocol receives a data packet for transmission, it first check whether the packet belongs to an already established flow by matching the source node address and the flow's unique identifier with the same fields in $T_{Forward}$. If a match exists, the packet is forwarded to a downstream node whose address is present in the corresponding record of $T_{Forward}$. Otherwise, the forwarding module checks the C_{Table} for the route towards the sink node address present in the data packet header. If the route towards the sink node is available, the forwarding module adds a new record in $T_{Forward}$ for the new flow. In this way, it is possible that different flows originating from the same sink node may use different different paths towards the same sink node.

To illustrate that the routing protocol is capable of discovering and using multiple data forwarding paths towards the same sink node at different flows level, please consider the following example. Suppose there is a source node A and a sink node S in a network, and at time T_0 a new flow originates from A, and P_1 is the best path in-terms of end-to-end available bandwidth available in C_{Table} stored at A. As this is a new flow, therefore the forwarding module of the routing protocol adds a new record in $T_{Forward}$ stored at A with the path set to P_1. At time $T_{(0+\epsilon)}$, A discovers a new route with better end-to-end available bandwidth towards S, A replaces the previous route with the new better route in C_{Table}, and we call this routing path as P_2. As per the routing protocol design, the existing flow does not use P_2, as the flow's data are forwarded using $T_{Forward}$, and record corresponding to the existing flow in $T_{Forward}$ contains P_1. We further suppose that at time $T_{(0+\zeta)}$ where $(\zeta > \epsilon)$ a new flow emerges at A towards S. Before forwarding the new flow's data packets, the forwarding module of the routing protocol adds a new record for the new flow in $T_{Forward}$ with path set to P_2 (as when the new flow starts, the forwarding module does not find a path for the new flow in $T_{Forward}$, therefore the forwarding module takes P_2 from C_{Table}). In this way, ABR supports multipath routing at different flows level.

3.2.2 Route Repair

A node n in set N infers a route failure if the node does not receive a HELLO message from a downstream node for a predefined interval of time. In case of a route failure, node n tries to repair the route locally, i.e., by checking C_{Table} for an alternate data forwarding path. If there is no alternate path in C_{Table}, node n informs the upstream node (upstream node's address is stored in $T_{Forward}$) about the route failure, using a route failure message. The upstream node tries to repair the route locally, if unsuccessful it informs its upstream node about the route failure. This process continues until a node on $(k_{c,j})_n$ finds an alternate route, or the source node is informed about the route failure.

3.2.3 Loop-Free Routing

The ensure loop-free routing, the routing protocol uses sink sequence numbers. Loops even if they are short-lived can be harmful. If a new flow appears in a network, and at that time there was a loop on the best available data forwarding path, the new flow's data will not reach the sink node, because as per the protocol the new flow can only change the path in case of a route failure. Let us consider the following example to illustrate how sink sequence numbers can help to avoid loops.

Let us consider a line topology of five nodes $1 \leftrightarrow 2 \leftrightarrow 3 \leftrightarrow 4 \leftrightarrow 5$. Suppose that flow X starts from node 2 and terminates at node 5. After some time flow Y starts from node 4 and terminates

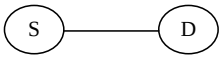
at node 5. As per the routing algorithm design, nodes within the interference range of node 4 update the available bandwidth towards node 5. We suppose that the updated information does not reach node 1, and node 1 advertised old higher available bandwidth information towards node 5. Therefore, node 2 thinks it discovered an alternate data forwarding path towards node 5 with higher end-to-end available bandwidth, hence it adds the new route in its C_{Table}. If now node 2 wants to start flow Z towards node 5, it forwards data to node 1 and node 1 transfers data to node 2, hence a loop is created. If sink sequence numbers are used, node 1 would have advertised the old information with a sink sequence number that node 2 already knows, as node 1 can only get node's 5 information from node 2. Therefore in our available-bandwidth-based routing protocol, node 2 only accepts the sink node information if the received sink sequence number is greater than the one stored at node 2. Hence in this example, node 2 avoids the loop.

3.3 Available Bandwidth and Contention-Aware Routing Protocol

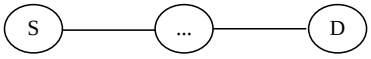
Using end-to-end available bandwidth as a routing metric may result in longer data forwarding paths. A longer path results in a higher intra-flow contention count [3]. Due to the higher intra-flow contention count, it is possible that the actual available bandwidth on a longer path is less than the actual available bandwidth on a comparatively shorter path. To demonstrate this point, let us consider the following example.

We suppose two data forwarding paths namely: path 1 and path 2. $1 \leftrightarrow 2 \leftrightarrow 3 \leftrightarrow 4$ is path 1, and $1 \leftrightarrow 5 \leftrightarrow 6 \leftrightarrow 7 \leftrightarrow 4$ is path 2. Node 1 is the source node and node 4 is the sink node. The minimum end-to-end available bandwidth on path 1 and path 2 is X and Y bps respectively, and (Y > X). We further suppose that the nodes within a two hop distance can cause interference. Therefore, the maximum intra-flow contention count on path 1 is at node 2, and it is equal to three times the data generation rate of node 1 (because the node cannot transmit when nodes 1 and 3 are transmitting, and the node also transmits data originating from node 1). Similarly, the maximum intra-flow contention count on path 2 is at nodes 5 and 6, and it is equal to four times the data generation rate of node 1. Further suppose that the minimum end-to-end available bandwidth on path 1 is equal to the net bandwidth available at node 2, i.e, B_2, and minimum end-to-end available bandwidth on path 2 is equal to the net available bandwidth at nodes 5 and 6, i.e., B_5 and B_6 respectively. Let us further suppose $\left(\frac{X}{3}\right) > \left(\frac{Y}{4}\right)$. In the given example, our ABR protocol would have selected the longer path (path 2) with lower actual end-to-end available bandwidth. Hence, we present the ABCR protocol, as in this example, it would have selected the shorter path (path 1) with the higher end-to-end available bandwidth.

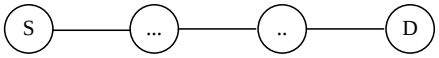
The maximum intra-flow contention ($CC_{n \to c}$) towards a sink node c in set C, can be determined using the algorithm presented in [3], and Figure 3 and Figure 4 summarize the algorithm. Nodes running ABCR advertise their hop-count distance towards each discovered sink node ($dist_{n \to c}$) using the HELLO message. As per the algorithm and as also shown in [23], ($CC_{n \to c}$) at any node is not more than 5, and intra-flow contention on any node depends on the node's hop-count distance from a source node and a sink node. For any node, it is really difficult to predict the source nodes whose flow(s) will traverse the node, therefore it is really hard to take into account the actual intra-flow contention count especially for a proactive routing protocol. ABCR approximate the intra-flow contention count using the node's hop-count distance from the sink node, and afterwards approximates the actual end-to-end available bandwidth at node n on any data forwarding path towards a sink node c, and it is equal to $AB_{n \to c}$. $AB_{n \to c} = \min(\frac{B_n}{\min(5, dist_{n \to c})}, B_{(k_{c,j})_{n-1}})$. In our ABCR protocol, each node advertises $AB_{n \to c}$ in the HELLO message, and C_{Table} maintains $AB_{n \to c}$ corresponding to each discovered sink node. Other design and working details of our ABCR protocol are the same as of our ABR protocol.



(a) Source and destination nodes are direct neighbors of each other. Contention factor at source and destination nodes is 1



(b) Destination node is two hop away from the source node. Contention factor at source and destination nodes is 2



(c) Destination node is more than two hops away from the source node. Contention factor at source and destination nodes is 3

Figure 3: Contention Factor on Source and Destination Node

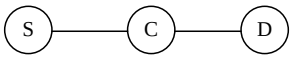
3.4 Shortest Hop-Count and Available-bandwidth-based Opportunistic Routing Protocol

The ABOR protocol tries to discover balanced multiple data forwarding paths (shortest paths with the same hop-count) towards each sink node, and afterwards ABOR selects a relaying node r from a set of downstream nodes based on the downstream nodes' available bandwidth. The following sub-section describes ABOR in more detail.

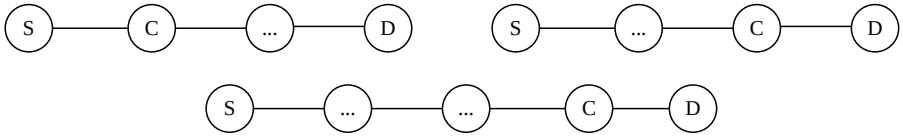
3.4.1 Data Forwarding Path Discovery and Maintenance

Each sink node c in set C broadcasts the same HELLO message that is discussed in Section 3.1, along with the following additional information: sink sequence number and shortest hop-count distance towards the sink node (in this case it is 0, as the message is being broadcasted by the sink node itself). A sink node increments the sink sequence number each time the sink broadcasts the HELLO message. Each node n in set N maintains a sink table C_{Table}, and each record in C_{Table} represents a single discovered sink node at node n. A single record in C_{Table} stores the following information: sink node address, sink sequence number, shortest hop-count distance towards the sink node, and a list of candidate downstream nodes providing the same shortest hop-count distance towards the sink node. The available bandwidth estimation algorithm discussed in Section 3.1 maintains the direct neighbours table, and the available bandwidth at each candidate downstream node is available in the direct neighbour table. Each node n in set N also broadcasts the HELLO message, and the HELLO message contains the information about the sink node(s) that the node n has discovered along with the information required by the available bandwidth estimator.

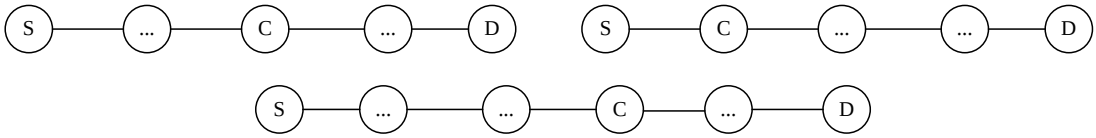
When any node n in set N receives the HELLO message, the node extracts the information about sink node(s) from the HELLO message. For each sink node information extracted from the HELLO message, the node compares the sink node address in C_{Table}. If the node does not find a match, the node adds a new record corresponding to the sink node in C_{Table}. If the node finds a match, the node compares the sink sequence number present in C_{Table} with the extracted sink sequence number. If the extracted sink sequence number is less than or equal to the sink sequence number present in C_{Table}, the extracted sink information is ignored. If the extracted sink sequence number is greater than the sink sequence number present in C_{Table}, the node compares the extracted shortest hop-count distance towards the sink node with the shortest hop-count distance towards the sink node present in C_{Table}. If the extracted shortest hop-count distance towards the sink node is greater than the shortest hop-count distance towards the sink node present in C_{Table}, the extracted sink node information is ignored. If the extracted shortest hop-count distance towards the sink node is equal to the shortest hop-count distance towards the sink node present in C_{Table}, the node adds the HELLO message broadcasting node to the



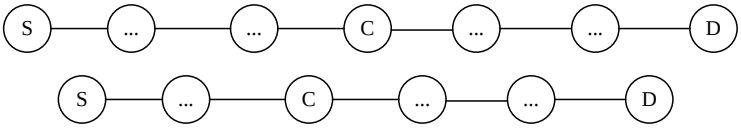
(a) Source and destination nodes are direct neighbors of the node calculating intra-flow contention factor, intra-flow contention factor at node C is 2



(b) Any one among the source or the destination node is direct neighbor of the node calculating intra-flow contention, and the other node is exactly two hops away from the node, or source node is more than two hops away, but the destination node is direct neighbor of the node then intra-flow contention factor is 3 at node C



(c) Node calculating intra-flow contention factor is exactly two hops away from the source and the destination node, or the node is direct neighbor of the source node, but the destination node is more than two hops away, or the source node is more than two hops away from the node, but the destination node is exactly two hops away from the node. In this case, the intra-flow contention factor at node C is 4



(d) Node calculating intra-flow contention is more than two hops away from both source and destination nodes, or source node is exactly two hops away from the node, and destination node is more than two hops away from the node. In this case intra-contention factor is 5

In all above scenario, node C is the node that is calculating the intra-flow contention

Figure 4: Intra-flow Contention Factor Estimation at Intermediate Relaying Nodes

list of candidate downstream nodes that provide the shortest hop-count distance towards the sink node, if the HELLO message broadcasting node is not already present in the candidate downstream list. If the extracted shortest hop-count distance towards the sink node is less than the shortest hop-count distance towards the sink node present in C_{Table}, the node deletes all the nodes in the candidate downstream nodes list, and adds the HELLO message broadcasting node in the candidate downstream nodes list. If a node n does not receive the HELLO message from a node that is in any of the sinks' candidate downstream list, within a pre-defined time interval, the node is removed from every candidate downstream list in which the node is present. Algorithm 2 summarizes the HELLO message processing, and in the algorithm rec_i represents a record corresponding to a sink node in the HELLO message.

If a node n receives a packet for transmission towards a sink node c, the node checks for the available candidate downstream node(s) present in C_{Table}. If a list of candidate downstream nodes is present in C_{Table}, the node forwards the packet to the candidate downstream node with highest available bandwidth B_n. B_n is available in the direct neighbour table. This process

continues till the packet is delivered to the sink node.

Algorithm 2: ABOR HELLO Message Processing

```
1  sink_record = Null;
2  downstream_node = Null;
3  for ∀rec_i ∈ hello_msg do
4  |  sink_record = is_sink_in_C_Table(rec_i.addr);
5  |  if sink_record == Null then
6  |  |  add_new_record_in_C_Table(rec_i);
7  |  end
8  |  else
9  |  |  if sink_record.seq_no > rec_i.seq_no then
10 |  |  |  continue;
11 |  |  end
12 |  |  else
13 |  |  |  if sink_record.hop_count < hello_msg.hop_count then
14 |  |  |  |  continue;
15 |  |  |  end
16 |  |  |  else
17 |  |  |  |  if sink_record.hop_count == hello_msg.hop_count then
18 |  |  |  |  |  downstream_node = is_node_in_downstream_list(hello_msg.src_addr);
19 |  |  |  |  |  if downstream_node == Null then
20 |  |  |  |  |  |  add_node_to_downstream_list(hello_msg.src_addr);
21 |  |  |  |  |  end
22 |  |  |  |  end
23 |  |  |  |  else
24 |  |  |  |  |  if sink_record.hop_count > hello_msg.hop_count then
25 |  |  |  |  |  |  update_record_in_C_Table(sink_record, rec_i);
26 |  |  |  |  |  end
27 |  |  |  |  end
28 |  |  |  |  update_seq_no_in_C_Table(sink_record, rec_i.seq_no);
29 |  |  |  end
30 |  |  end
31 |  end
32 end
```

3.4.2 Route Repair

A node n in set N infers a route failure towards a sink node c, if for a predefined interval of time, the node does not receive the HELLO message from the last available downstream node present in the corresponding record (record pertaining to the sink node c) of the candidate downstream nodes list (please note that in case of route failure, the node keeps both the sink node address and the sink sequence number in C_{Table} for a pre-defined interval of time, this helps to avoid routing loops). In this case, the node broadcasts a route failure message, the route failure message contains the address of the sink node that is no longer reachable through the node. Upon receiving the route failure message, upstream node(s) removes the node from the corresponding record of the sink table, if present. If the node was the last available downstream

Table 1: General Simulation Parameters

Parameter	Value
MAC layer	Unslotted CSMA-CA
MAC layer reliability	Enabled
Radio duty cycling algorithm	No radio duty cycling
Radio model	Unit disk graph model
MAC layer queue size	30 frames
Channel rate	250 kbps
Node transmission range	50 meters
Node carrier sensing range	100 meters
Total frame size	127 bytes
Emulated mote	Tmote Sky

node in the corresponding record of the candidate downstream nodes list of any of the upstream nodes, the same process is repeated.

3.4.3 Loop-Free Routing

The way the sink sequence number is used by our ABOR protocol helps to create loop-free forwarding paths. Let us consider the following example.

Let us consider a line topology of five nodes $1 \leftrightarrow 2 \leftrightarrow 3 \leftrightarrow 4 \leftrightarrow 5$. Suppose that node 1 is the source node, and node 5 is the sink node. Using our routing protocol, nodes 1, 2, 3, and 4 discover that they can reach the sink node (node 5) in 4, 3, 2, and 1 hops respectively. We further suppose that the link between node 3 and 4 fails, hence node 3 does not have a valid route towards the sink node. After some time, node 3 receives the HELLO message from node 2, and node 3 discovers that it can reach the sink node in 3 hops using node 2. But, after comparing the received sink sequence number with the sink sequence number stored in its C_{Table}, node 3 discards the sink information received from node 2, as the sink sequence number is already known to node 3 (node 2 in this case can only get the sink sequence number updates from node 3). Therefore, the forwarding loop (node 3 $\leftrightarrow$ node 2 $\leftrightarrow$ node 3) is avoided.

4 Simulation Results

We performed extensive simulation experiments to thoroughly evaluate the available bandwidth as a routing metric. Simulations were performed using the widely used Cooja WSN simulator that uses real programming code for a wireless sensor node [24]. In our simulations we used a grid network topology, and the network spans an area of $300 \times 300\, m^2$. We varied the number of nodes in a network, and 6 nodes are randomly selected as source nodes. Bandwidth is a shared and scarce resource in IEEE 802.15.4-based networks, therefore we opted for a total of 6 source nodes. To study the impact of multiple sink nodes on the performance of available-bandwidth-based routing metrics, we incrementally increase the number of sink nodes from 1 to 4, and the sink nodes are randomly placed. In case of multiple sink nodes, each source node randomly selects a sink node for its flow. Table 1 shows general simulation parameters. Our results are based on 10 simulations for each number of sink nodes (randomly selecting the source nodes and data generation rates each time). In the figures, we plot the mean value for each protocol, and we show as error bars the 95% CI around the mean, based on t-distribution with a sample size of 10. In this section where CIs overlap and means are not in the overlap region, we base our

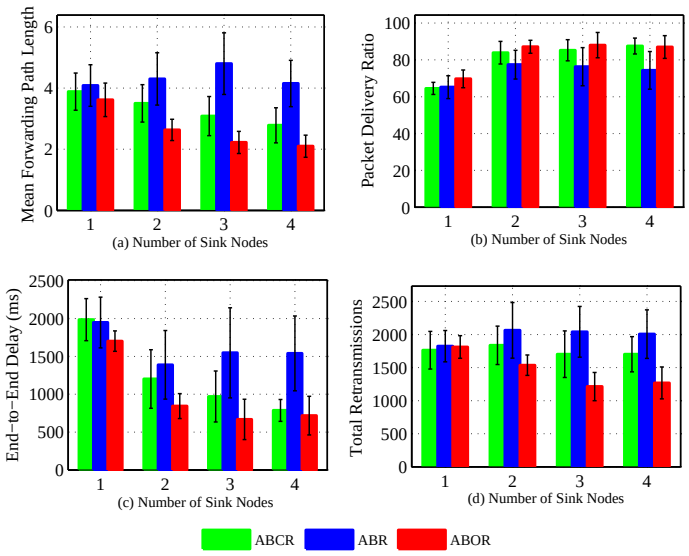


Figure 5: Routing Protocols' Performance Comparison (50 Nodes Network)

conclusions on the results of a t-test.

In this section, we report the mean PDR and mean end-to-end delay because they are important QoS metrics. The mean data forwarding path length may impact PDR and end-to-end delay, therefore we also present results about the mean path length. A higher number of data packets retransmissions can be an indication of congestion along a path. Therefore, collecting data about the number of retransmitted data packets can give an indication about a routing protocol's ability to avoid congested paths (excluding situations when the data packet retransmission is due to a route failure). For the stated reason, we also report the mean total number of retransmissions.

4.1 Flow-based Data Traffic

Each source node's flow data generation rate is randomly distributed in the range [4, 8] kbps. The source nodes start data transmission at 10, 20, 30, 40, 50, and 60 simulation seconds respectively. All source nodes terminate their flows at 100 simulation seconds, and the total duration of each simulation is 115 seconds.

Figure 5 shows comparison of our different available-bandwidth-based routing protocols. In case of a single sink node, the routing protocols demonstrate similar performance in terms of mean path length, mean PDR, mean end-to-end delay, and mean total number of retransmissions. Convergence of all flows' data packets near the sink node increases contention near the sink, and increased contention results in a similar performance. In case of multiple sink nodes, ABOR demonstrates better performance compared to ABR, and in some cases it also demonstrates better performance compared to ABCR. Among the other two routing protocols, in some cases ABCR demonstrates better performance. Generally, the results demonstrate that an available-bandwidth-based routing protocol that selects lengthy data forwarding paths suffers from higher intra-flow contention, and the higher intra-flow contention negatively impacts the performance.

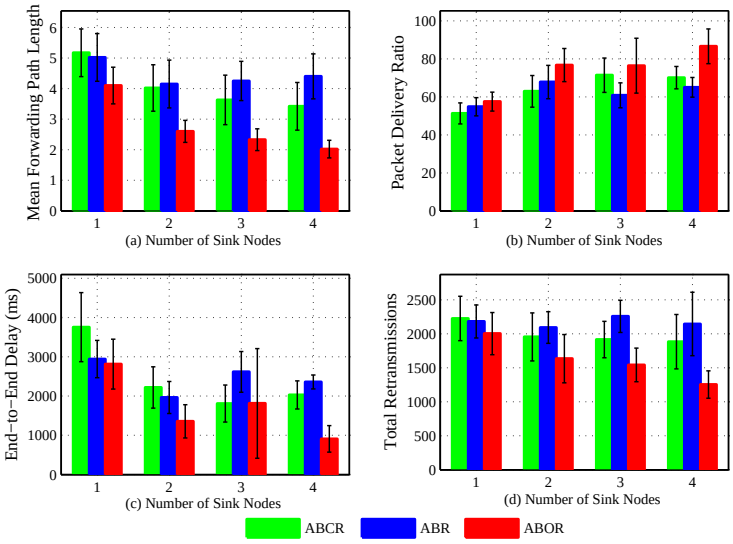


Figure 6: Routing Protocols' Performance Comparison (75 Nodes Network)

Figure 6 shows the same measures as Figure 5, but for a 75 node network. In general, Figure 6 and Figure 5 show similar trends, i.e., ABOR demonstrates better performance relative to the other routing protocols. In all cases, ABR and ABCR demonstrated similar performance, and this is the only difference compared to the results shown in Figure 5. It can be observed in this case as well that lengthy data forwarding paths negatively impact the performance of an available-bandwidth-based routing protocol.

Figure 7 shows the same measures for a 100 node network. The figure demonstrate a key difference compared to Figure 5 and Figure 6, i.e., ABOR selects shorter data forwarding paths compared to the other routing protocols, but it demonstrates lower PDR compared to ABCR. The difference is due to the fact that ABOR results in a higher total number of retransmission on average. The higher total number of retransmissions is an indication that ABOR selects congested paths. Hence, congestion on the selected paths results in the inferior performance. ABR demonstrates similar performance trend as for 50 node and 75 node networks. The results shown in Figure 7 demonstrate that not only path length, but congestion on the selected path also impacts the performance of available-bandwidth-based routing protocol.

Figure 8(a) shows the protocols' per-node overhead. The protocols use the same HELLO and one-hop neighbours information messages, therefore the protocols demonstrate the same overhead. In case of 50 and 75 node networks, the per-node control overhead is 1.739 kbps, and for the 100 node network, the overhead is 1.87 kbps. As the number of nodes in a network increases, the mean number of direct neighbours of a node increases, therefore the node may broadcast multiple one-hop neighbour information messages per time unit. This results in increased control overhead, as demonstrated by the per-node overhead for the 100 node network. Figure 8(b) shows the network-wide protocols' total control overhead for the total duration of the simulation. The overhead increases as the number of nodes increases.

ABOR uses shortest hop-count to build a set of candidate downstream nodes towards a particular sink node. ABOR forwards data packets to the candidate downstream node with the highest available bandwidth among all the candidate downstream nodes, hence ABOR is

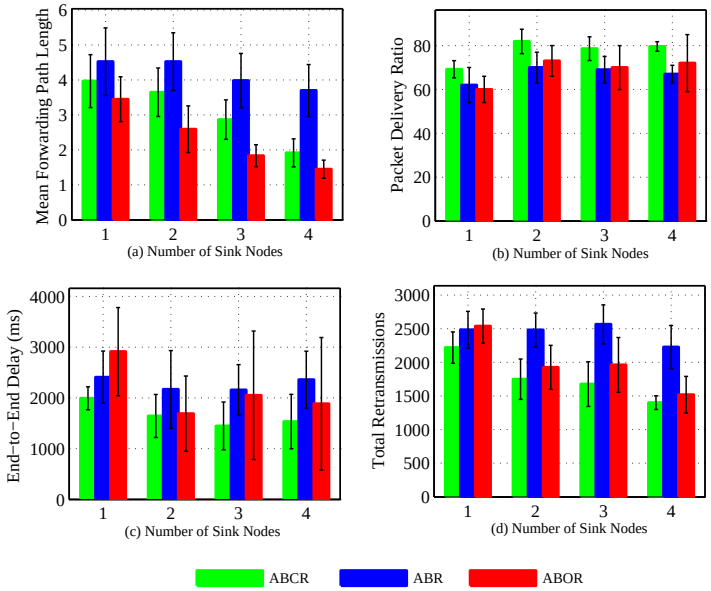


Figure 7: Routing Protocols' Performance Comparison (100 Nodes Network)

capable of distributing the same flow's data packets on multiple data forwarding paths. There were scenarios where the approach has shown good results. ABR and ABCR do not distribute the same flow's data packet on multiple paths. Therefore, we modified ABR and ABCR such that both routing protocols maintain their top K downstream nodes (the downstream nodes that advertised best paths w.r.t. the routing metric being used) corresponding to each sink in a routing table. The modified ABR and ABCR protocols do not maintain a separate forwarding table for established flows. At each relaying node, the routing protocols' forwarding modules distribute data packets of a flow by using the top K downstream nodes in a round-robin manner, i.e., the first data packet is being forwarded to the first downstream node, the second data packet is being forwarded to the second downstream node, and K^{th} data packet is forwarded to the K^{th} downstream node, and then the cycle restarts. If a node discovers a new better path

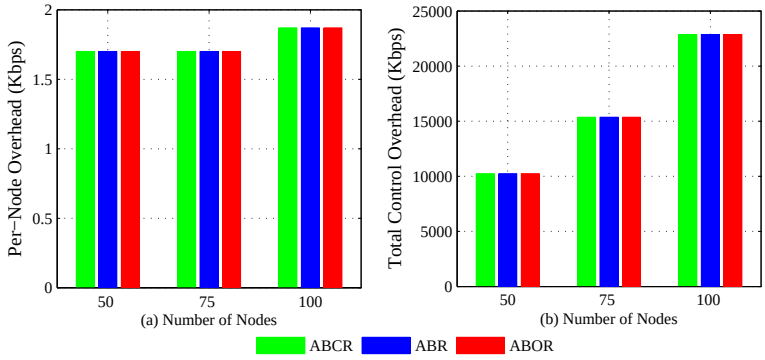


Figure 8: Routing Protocols' Control Overhead

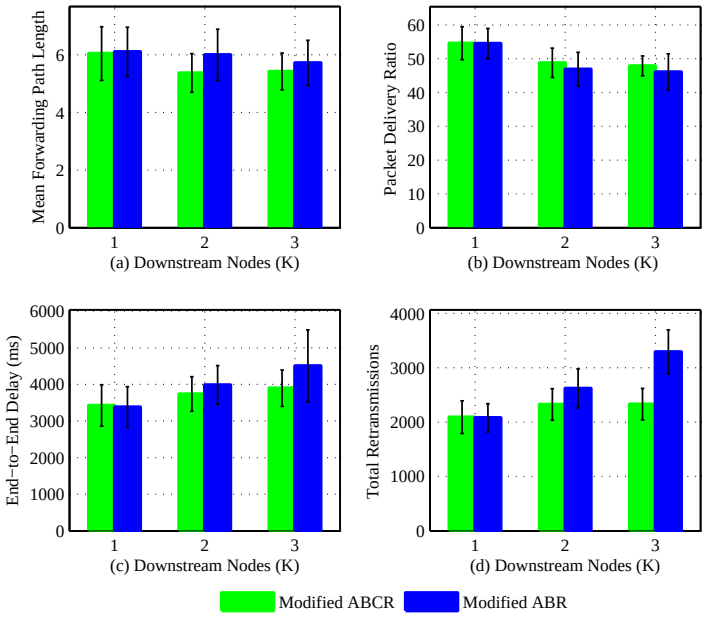


Figure 9: Routing Protocols' Performance Comparison (1-Sink Scenario)

towards a particular sink node through any of its downstream nodes, and the number of already available downstream nodes corresponding to the sink node in the node's routing table is equal to K, the routing protocols replace the worst available downstream node (the downstream node corresponding to the worst data forwarding path) with the new downstream node. If the number of downstream nodes corresponding to the sink record is less than K, another downstream node is added to the downstream node list of the sink node inside the node's routing table. Afterwards, already existing and new data flow(s) start using the newly discovered path as well. The other working details of the modified ABR and ABCR protocols are the same as for the original ABR and ABCR protocols respectively.

For performance evaluation of the modified ABR and ABCR protocols we choose a 75 node network, because in the performance evaluation of ABR, ABCR, and ABOR, both ABR and ABCR protocols demonstrated slightly inferior performance compared to their performance in other node densities. Therefore, we assume that if modified ABR and ABCR demonstrate better performance in the 75 node network, there is a higher probability that both routing protocols will demonstrate better performance in other node densities as well. Moreover, we created two set of simulation scenarios: 1-sink scenario and 4-sinks scenario. We consider the 1-sink scenario as a worst-case scenario because in this scenario all data traffic merges near the single sink node. We consider the 4-sinks scenario as a best-case scenario because if each source node randomly selects a sink node for its flow, there are good chances that all data packets do not merge at a single location. Each simulation is repeated 10 times, and other simulation details are the same as were for ABR, ABCR, and ABOR performance evaluation.

For K equal 1, the modified ABR and ABCR protocols always change a flow's forwarding path to the best available path, however the original ABR and ABCR protocols fix the flow's path. Figures 9(a), 9(c), and 9(d) demonstrate that for a value of K equal 1, on average ABCR demonstrated little improvement in mean PDR, mean end-to-end delay, and mean total number

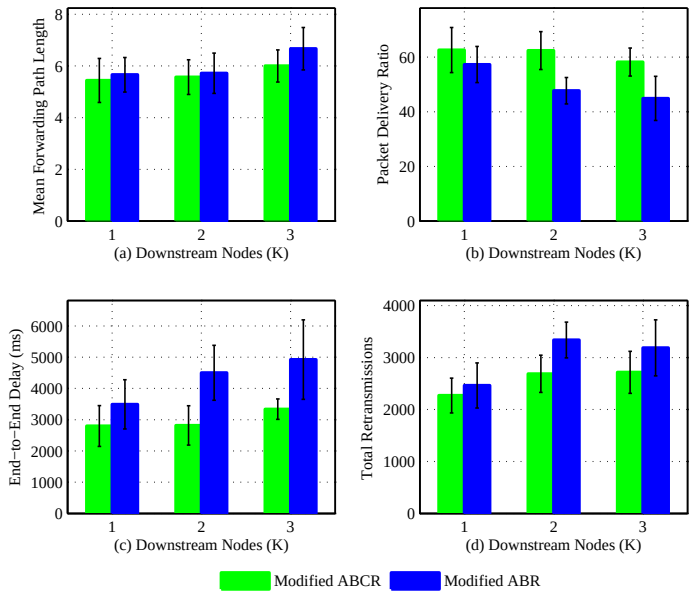


Figure 10: Routing Protocols' Performance Comparison (4-Sinks Scenario)

of retransmissions compared to the original ABCR protocol (Figure 6). But, for the same value of K, on average ABR demonstrates inferior performance compared to the original ABR routing protocol as shown in Figure 6. This is because of the fact that in this case modified ABR on average selects lengthy data forwarding paths compared to the original ABR protocol. With an increase in the value of K on average the performance of the modified ABCR and ABR protocols deteriorates. This was due to the fact that instead of using the best path for different data packets, the data packets are also forwarded on different less optimal paths as well. On a statistical basis, both routing protocols demonstrate similar performance most of the time. Overall, maintaining K top downstream nodes for data distribution using the modified ABR and ABCR protocols does not show an improvement compared to the ABR and ABCR protocols. On Average ABR and ABCR demonstrated better performance compared to modified ABR and ABCR most of the time.

Figure 10 demonstrates that with an increase in the value of K for both routing protocols the mean forwarding path length, mean end-to-end delay, and total number of retransmissions increase, whereas PDR drops. In contrast to the 1-sink node scenario, modified ABCR demonstrates higher PDR and lower mean end-to-end delay compared to modified ABR for values of K greater than 1. Both routing protocol selects paths of similar lengths, and demonstrate similar total number of retransmissions. Again, modified ABR and ABCR demonstrated poor performance compared to the performance of ABR and ABCR protocols on average. The reasons for the poor performance are the same as given for 1-sink node scenario.

4.2 Event-based Data Traffic

To evaluate the performance of ABCR, ABR, and ABOR in event-driven IEEE 802.15.4-based ad-hoc networks, we performed another set of simulation experiments. For the experiments, we used the 50 node network. Initially, 6 nodes randomly detect an event between [5, 15] seconds.

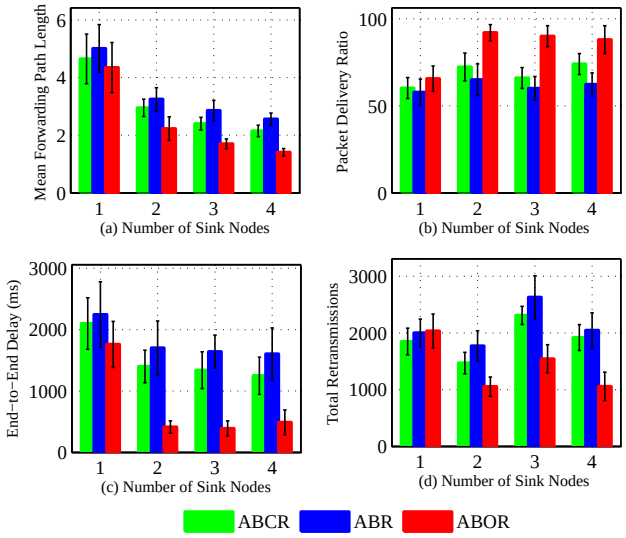


Figure 11: Routing Protocols' Performance Comparison (Event-based Data Traffic)

Table 2: ABR Results (10 Frames/Second)

Min. Avb. Bandwidth (kbps)		**PDR (%)**		**MAC Overhead (kbps)**	
Mean	**CI**	**Mean**	**CI**	**Mean**	**CI**
6.770	3.262 - 1.0278	98.55	97.92 - 99.18	197	183 - 211

Once a node detects an event, the nodes within the three hop distance of the node also detect the same event. After detecting an event, the nodes transmit 2 packets per second for 10 seconds. Once an event is detected the nodes can randomly detect another event after [20, 25] seconds. No node detect events after 90 seconds, and each simulation terminates after 115 seconds. Our traffic generation model is a representation of a data traffic generated by a range of event-detection system, e.g., fire detection, target tracking, etc. Other simulation parameters are the same as used in our flow-based data traffic experiments.

Figure 11 compares our different available-bandwidth-based routing protocols using an event-driven data generation pattern. In case of a single sink, the routing protocols demonstrate similar performance in terms of mean path length, mean PDR, mean delay, and mean total number of retransmissions. In case of multiple sinks, ABOR demonstrates better performance compared to the other two protocols. In some cases, among the other two protocols, ABCR demonstrates better performance. Mostly, the results shown in Figure 11 are consistent with the results shown in Figure 5. Therefore, the results demonstrate that a lengthy forwarding path not only negatively affects the performance of the available-bandwidth-based routing protocols in flow-based data traffic, but it also affects the performance of the protocols in event-based traffic.

4.3 Stress Tests

ABR selects single data forwarding paths using the end-to-end available bandwidth routing metric, whereas ABCR also considers the impact of intra-flow contention on a path's end-to-end available bandwidth. Therefore, we conduct stress tests, i.e., if a routing protocol discovers

Table 3: ABCR Results (10 Frames/Second)

Min. Avb. Bandwidth (kbps)		PDR (%)		MAC Overhead (kbps)	
Mean	**CI**	**Mean**	**CI**	**Mean**	**CI**
1.666	0.979 - 2.353	97.1	96.06 - 98.13	226	214 - 238

Table 4: ABR Stress Test Results (16 Frames/Second)

Min. Avb. Bandwidth (kbps)		PDR (%)		MAC Overhead (kbps)	
Mean	**CI**	**Mean**	**CI**	**Mean**	**CI**
0	0 - 0	74.7	71.96 - 77.44	259	246 - 272

that X bps are available on a path, a source transmits Y bps (ideally equal to X bps) on the path and monitors the following: minimum available bandwidth on the path, PDR, and MAC layer overhead (back-off, ACK waiting time, ACK transmissions, and retransmissions). For the tests, we use the 50 node network. There is a single source and sink in the network. The source transmits 10 data frames per second, and the size of each frame is 127 bytes. Therefore, the source transmits approximately 10 kbps. The source starts its transmission at 10 seconds, and terminates the transmission at 100 seconds. Each simulation experiment terminates at 115 seconds. We do not conduct stress test for ABOR as it does not use a single path.

Table 2 and Table 3 show mean minimum available bandwidth, PDR, and MAC layer overhead (accumulative overhead of nodes transmitting data packets) corresponding to ABR and ABCR respectively on the selected data forwarding path. As ABCR considers the impact of intra-flow contention on the path's available bandwidth, it reports statistically significantly lower available bandwidth compared to ABR. Moreover, both protocols demonstrate similar PDR. Both protocols report some available bandwidth, however the PDR is not 100%, some frames were dropped at the MAC layer because the retransmission attempts threshold was reached and ACKs for those frames were not received. As ABR and ABCR report 6.770 kbps and 1.666 kbps available bandwidth respectively, to perform the stress test, we increase the number of data frames to 16 frames/second and 12 frames/second corresponding to ABR and ABCR respectively. Table 4 and Table 5 show the stress test results. With the increase in the data traffic ABR and ABCR report 0 bps available bandwidth. ABR demonstrates substantially lower PDR compared to its PDR shown in Table 2. ABCR also demonstrates lower PDR compared to its PDR shown in Table 3, but its PDR is higher than ABR. Thus, ABCR's estimate of the available bandwidth on a path is more realistic compared to ABR's estimate. Results shown in Tables 2, 3, 4, and 5 demonstrate that with an increase in the data traffic, the MAC layer overhead also increases. Therefore, from these results we conclude that among others intra-flow contention and increase in the MAC layer overhead with an increase in the data traffic impact the available bandwidth. The increase in the MAC layer overhead at a node depends on the future data traffic within the interference range of the node. It is not possible for a routing protocol to exactly estimate the future data traffic, therefore our protocols do not consider the impact of future MAC layer overhead on the available bandwidth.

Table 5: ABCR Stress Test Results (12 Frames/Second)

Min. Avb. Bandwidth (kbps)		PDR (%)		MAC Overhead (kbps)	
Mean	**CI**	**Mean**	**CI**	**Mean**	**CI**
0	0 - 0	86.5	83.09 - 89.91	261	251 - 271

5 Conclusions and Future work

We extensively evaluated the available bandwidth as a routing metric for delay-sensitive IEEE 802.15.4-based ad-hoc networks. For the purpose of the evaluation, we not only used end-to-end available bandwidth as a routing metric, but combined the available bandwidth with other routing metrics, i.e., shortest hop-count and intra-flow contention count. The different routing metrics presented in this paper were used to design both single best path and multi-path routing protocols for both single-sink and multi-sink networks. Our experimental analysis shed light on the limits of the available bandwidth as a routing metric in IEEE 802.15.4-based ad-hoc networks. The results presented in this paper demonstrated that available-bandwidth-based routing metric shows better results when combined with other routing metrics, i.e., shortest hop-count and intra-flow contention-count. Moreover, for multi-path data forwarding towards the same sink node, and at high traffic volumes an available-bandwidth-based routing metric performs best when combined with the shortest hop-count routing metric. Evaluating the available bandwidth as a routing metric on a real IEEE 802.15.4-based ad-hoc networks is our future work.

References

[1] T. O'donovan, J. Brown, F. Büsching, A. Cardoso, J. Cecílio, J. Do, P. Furtado, P. Gil, A. Jugel, W.B. Pöttner, U. Roedig, J. Silva, R. Silva, C. J. Sreenan, V. Vassiliou, T. Voigt, L. Wolf, and Z. Zinonos. The GINSENG System for Wireless Monitoring and Control: Design and Deployment Experiences. *ACM Transactions on Sensor Networks*, 10(1):4:1–4:40, December 2013.

[2] C. Sarr, C. Chaudet, G. Chelius, and I. G. Lassous. Bandwidth Estimation for IEEE 802.11-based Ad Hoc networks. *IEEE Transactions on Mobile Computing*, 7(10):1228–1241, 2008.

[3] M. O. Farooq and T. Kunz. BandEst: Measurement-based Avaialble Bandwidth Estimation and Flow Admission Control Algorithm for Ad-Hoc IEEE 802.15.4-based Wireless Multimedia Networks. *International Journal of Distributed Sensor Networks*, 2015:1–15, 2015. Article ID: 539048.

[4] H. Alwan and A. Agarwal. Reliable Fault-Tolerant Multipath Routing Protocol for Wireless Sensor Networks. In 25^{th} *Biennial Symposium on Communication*, pages 323–326, 2010.

[5] H. Alwan and A. Agarwal. Multi-Objective Reliable Multipath Routing for Wireless Sensor Networks. In *Globecom workshops*, pages 1227–1231, 2010.

[6] L. Cheng, J. Niu, J. Cao, S. K. Das, and Y. Gu. QoS Aware Geographic Opportunistic Routing in Wireless Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems*, 25(7):1864–1875, 2014.

[7] Y. Yim, H. Park, J.l Lee, S. Oh, and S. Kim. Distributed Forwarder Selection for Beaconless Real-Time Routing in Wireless Sensor Networks. In *IEEE* 77^{th} *Vehicular Technology Conference*, pages 1–5, 2013.

[8] L. Cheng, J. Cao, C. Chen, J. Ma, and S. K. Das. Exploiting Geographic Opportunistic Routing for Soft QoS Provisioning in Wireless Sensor Networks. In *IEEE* 7^{th} *International Conference on Mobile Adhoc and Sensor Systems*, pages 292–301, 2010.

[9] P. Spachos, Liang Song, and D. Hatzinakos. Performance Comparison of Opportunistic Routing Schemes in Wireless Sensor Networks. In 9^{th} *Annual Communication Networks and Services Research Conferencel*, pages 271–277, 2011.

[10] D. Djenouri and I. Balasingham. Traffic-Differentiation-Based Modular QoS Localized Routing for Wireless Sensor Networks. *IEEE Transactions on Mobile Computing*, 10:797–809, 2011.

[11] M. Liu, S. Xu, and S. Sun. An Agent-assisted QoS-based Routing Algorithm for Wireless Sensor Networks. *Journal of Network and Computer Applications*, 35:29–36, 2012.

[12] J. Kim, H. Seo, and J. Kwak. Routing Protocol for Heterogeneous Hierarchical Wireless Multimedia Sensor Networks. *Wireless Personal Communications*, 60(3):559–569, 2011.

[13] H. Yoo, M. Shim, and D. Kim. A Scalable Multi-Sink Gradient-Based Routing Protocol for Traffic Load Balancing. *EURASIP Journal on Wireless Communication and Networking*, 2011(85):1–16, 2011.

[14] H. Liu, Z. L. Zhang, J. Srivastava, and V. Firoiu. PWave: A Multi-source Multi-sink Anycast Routing Framework for Wireless Sensor Networks. In *IFIP NETWORKING*, volume 4479 of *Lecture Notes in Computer Science*, pages 179–190. 2007.

[15] N. Mitton, D. Simplot-Ryl, M. E. Voge, and L. Zhang. Energy Efficient k-Anycast Routing in Multi-sink Wireless Networks with Guaranteed Delivery. In *Ad-hoc, Mobile, and Wireless Networks*, volume 7363 of *Lecture Notes in Computer Science*, pages 385–398. 2012.

[16] Y. Wu and W. Liu. Routing Protocol based on Genetic Algorithm for energy Harvesting Wireless Sensor Networks. *IET Wireless Sensor Systems*, 3:112–118, 2013.

[17] C. Park and I. Jung. Traffic-Aware Routing Protocol for Wireless Sensor Networks. In *International Conference on Information Science and Applications*, pages 1–8, 2010.

[18] P. Zhao, X. Yang, A. Ye, and S. Yang. joint multipath routing and admission control with bandwidth assurance for 802.11-based wmns. In *2011 IEEE Wireless Communications and Networking Conference*, pages 956–961, 2011.

[19] P. Zhao, X. Yang, J. Wang, B. Liu, and J. Wang. BOR/AC: Bandwidth-Aware Opportunistic Routing with Admission Control in Wireless 'Mesh Networks. In *IEEE INFOCOM*, pages 2701–2705, 2012.

[20] R. Hou, K. S. Lui, F. Baker, and J. Li. Hop-by-Hop Routing in Wireless Mesh Networks with Bandwidth Guarantees. *IEEE Transactions on Mobile Computing*, 11(2):264–277, 2012.

[21] R. Hou, K. S. Lui, and J. Li. Routing in Multi-Radio Multi-Channel Multi-Hop Wireless Mesh Networks with Bandwidth Guarantees. In *IEEE 73^{rd} Vehicular Technology Conference*, pages 1–5, 2011.

[22] M. O. Farooq and T. Kunz. Proactive Bandwidth Estimation for IEEE 802.15.4-based Networks. In *IEEE 77^{th} Vehicular Technology Conference*, pages 1–5, 2013.

[23] M. O. Farooq and T. Kunz. Key Factors for a Proper Available-Bandwidth-Based Flow Admission Control in Ad-Hoc Wireless Sensor Networks. In *Ad-hoc Networks and Wireless*, Lecture Notes in Computer Science, pages 246–260. Springer Berlin Heidelberg, 2015.

[24] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt. Cross-Level Sensor Network Simulation with Cooja. In *31^{st} IEEE Conference on Local Computer Networks*, pages 641–648, 2006.