Distributed Maintenance of Minimum-cost Path Information in Wireless Sensor Networks

Marcel Steine[1], Marc Geilen[1] and Twan Basten[1,2]
[1]Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands
[2]Embedded Systems Institute, Eindhoven, The Netherlands
{m.steine,m.c.w.geilen,a.a.basten}@tue.nl

ABSTRACT

The quality of the communication links in a Wireless Sensor Network often shows significant asymmetry and variation over time, due to, for example, heterogeneous settings of the transmission power, moving nodes or changing external interference. This makes it difficult for nodes to accurately maintain system-level properties, such as the minimum-energy path from the node to a given reference node, as required by many protocols. In this paper, we introduce a distributed service that allows nodes to maintain accurate information related to the minimum-cost path, such as its cost or parent on that path. Using controlled n-hop forwarding, to deal with asymmetric links, every node disseminates minimum-cost path and connectivity information allowing every connected node in the network to iteratively derive minimum-cost path information. This controlled n-hop forwarding is repeated to avoid stale information due to dynamic changes in link qualities. The parameters of the service allow a trade-off between the accuracy and overhead. We study the characteristics of a deployment that impact this trade-off and how the service should be parameterized accordingly. Extensive simulations and experiments for an actual deployment show a significant increase in the accuracy of the maintained minimum-cost path information, compared to the typically used local broadcasting approach.

Categories and Subject Descriptors: C.2.1 [Network Architecture and Designs]: Wireless communication; C.2.4 [Computer-Communication Networks]: Distributed Systems; C.4 [Performance of Systems]: Performance attributes.
General Terms: Algorithms, Performance.

1. INTRODUCTION

A Wireless Sensor Network (WSN) is a distributed network of small autonomous devices that are capable of sensing, processing and wireless communication. These devices, commonly called nodes, contain one or more sensors and are constrained by stringent resource limitations. They have to collaborate to perform a task specified by the end-user of the WSN. Typical examples can be found in the area of health care and environmental monitoring.

A WSN is a distributed system, but protocols often rely on knowledge of information pertaining to the entire network. A typical example of such a system-level property is a node's *minimum-cost path*, such as the minimum-energy path, towards a reference node. The cost can take any form such as the hop-count or expected energy or latency of sending a packet over the path. The cost of the minimum-cost path is directly used by routing and localization protocols [12, 15] to determine a node's location relative to others. Furthermore, the neighbouring node on the minimum-cost path closer to the destination is often used as a 'parent' to forward information to by data-collection and routing protocols [4, 6, 8, 17, 22].

For the correct and efficient operation of protocols, it is important that the minimum-cost path information maintained by the nodes is *accurate*, i.e., close to the 'real' value in the practical situation. For example Gradient Based Routing [15] depends on local knowledge about the minimum number of hops required to send packets to the sink. Packets are only forwarded if received from nodes with a higher hop-count. If nodes assume a hop-count that is higher than it is in reality, routing becomes suboptimal and the network load may increase. Deriving a lower hop-count can even potentially result in packets not reaching the sink at all.

In current work, distributed maintenance of minimum-cost path information is typically done in an ad-hoc manner as part of the protocol requiring the information. This is mostly done using a local broadcasting approach [4, 6, 8, 12, 15, 17, 21, 22]. The cost of the minimum-cost path and any additional information is broadcast by nodes which have this information. Starting from the node to which the minimum-cost path needs to be determined, information thereby floods through the network, allowing all (connected) nodes to infer minimum-cost path information in an iterative fashion. With this approach, nodes implicitly assume that they can send packets to every node from which they receive information without checking the existence and quality of the reverse link. Thus, this local broadcasting assumes symmetric links, i.e., bidirectional communication at the same cost. Furthermore, the broadcasting is often performed until the information converges to a single optimum, assuming that there is a fixed minimum-cost path with static cost over time. In practice, we see that link quality, for example expressed as a Packet Reception Ratio (PRR) [20], is often not symmetric and does vary [9, 23, 24]. Link quality changes,

PM^2HW^2N'11, October 31, 2011, Miami, Florida, USA.

for example, due to changing external interference, dynamic adaptation of parameters, such as the transmission power [10], or node mobility. Asymmetry exists, for example, due to the use of a (heterogeneous) set of nodes with different transmission capabilities.

In this paper, we introduce a distributed service that allows nodes to maintain accurate information about minimum-cost paths in the presence of links with a quality that can be asymmetric and vary over time. The maintained information can then be used for the accurate and efficient execution of any protocol in the protocol stack that depends on this information. The service disseminates both a node's minimum-cost path and connectivity information to allow other connected nodes in the network to derive their minimum-cost path information in an iterative fashion. In the presence of asymmetric links, controlled n-hop forwarding is used. The dissemination of information is then repeated to avoid stale information caused by dynamic changes in the link qualities. The design and parameters of the service allow an easy integration into existing and new WSN deployments and allow a trade-off between the accuracy of the information and the overhead of maintaining it. Using simulations and experiments, we provide insights in this trade-off and on how to set the parameters of the service for the required accuracy. We furthermore show that using the service results in a significant increase in the accuracy of path information and efficiency of a protocol using it, compared to the typically used local broadcasting approach.

The remainder of this paper is organized as follows. In Section 2, we discuss related work. Section 3 takes a closer look at the derivation of minimum-cost path information and introduces our service proposed to maintain this information on the nodes in a distributed manner. Section 4 discusses the performance of the service in more detail and shows simulation and experimental results. Section 5 concludes.

2. RELATED WORK

The minimum-cost path is an important concept used by many protocols. The most obvious use of the minimum-cost path is for routing and data collection protocols, where packets are forwarded to the 'best' neighbouring node(s) through which a sink can be reached. With Directed Diffusion [8] nodes determine their parent, or 'gradient', based on the rate of received interest packets. Gradient-Based Routing (GBR) [15] is a modified version of Directed Diffusion, where nodes record the number of hops to reach the sink and packets are only forwarded when received from a node with a higher hop-count. As energy-efficiency is important for WSNs, many routing protocols focus on energy-efficient routing [4, 17, 22] and use energy related factors, such as communication energy consumption rates and residual energy levels, to determine the cost of a path. Furthermore, the state-of-the-art data collection protocol Collection Tree Protocol [6] as used in TinyOS 2.x [19], depends on local knowledge of the minimum number of transmissions to reach a sink. Besides for many routing protocols, a minimum-cost path is also used in localization protocols, for example based on the popular 'DV-Hop' [12] protocol. This localization protocol derives the physical location of a node from the number of hops that the node is away from anchor nodes with a known location.

To determine the minimum-cost path for every node in the network, also referred to as setting up a cost field [21] or gradient field [6, 8, 22], in a distributed fashion all the above approaches require the nodes to use a local broadcast of their minimum-cost path information, also referred to as 'interests' [8] or 'advertisements' [22], as soon as it is locally known. This broadcasting is done multiple times until the minimum-cost path is found, thereby assuming the minimum-cost path cannot become worse over time. For example, with GBR nodes broadcast their hop-count such that nodes in an iterative fashion determine their hop-count based on the lowest hop-count received from neighbouring nodes.

Several solutions have been proposed to deal with specific problems related to asymmetric links in particular protocols outside the context of minimum-cost path derivation. The majority of protocols ignores them altogether [20]. It is desirable to exploit asymmetric links both for performance and for basic connectivity of wireless networks [14]. To be able to communicate in the presence of asymmetric links, a technique to relay [5] information around these asymmetric links, by using limited forwarding over multiple hops, can be used. As part of our service, we use a controlled version of this n-hop forwarding where already forwarded packets are not forwarded again to avoid extensive packet forwarding.

To the best of our knowledge, the impact of varying link qualities is not explicitly considered in current work on the distributed derivation of global information, such as the minimum-cost path. The typical assumption is that a node has a fixed minimum-cost path, i.e., the network is static, and the local information should converge to this. In many practical WSNs we cannot make this assumption and the minimum-cost information can become both better or worse over time. With our distributed service we explicitly consider the minimum-cost path not being stable by using a repeated update of information and we explore the related trade-off between accuracy and overhead.

We position our approach as a service, a commonly used abstraction for providing functionalities to other protocols, which can work independent of other protocols in the network stack. This relieves the protocol that requires minimum-cost path information from the non-trivial task of maintaining its own information. It also allows an easy integration in existing and new protocol stacks. Information can furthermore easily be shared in case multiple protocols in the stack require it, thereby avoiding the additional overhead caused by multiple protocols deriving the same information.

3. MAINTENANCE SERVICE

In this section, we introduce our service which lets nodes provide sufficient information to neighbouring nodes allowing all nodes to locally maintain global information related to the minimum-cost path in a distributed manner. In Section 3.1, we describe this goal in more detail with the use of an introductory example. In Section 3.2 we discuss how to communicate the required information between nodes in the presence of asymmetric links. Section 3.3 discusses how nodes can furthermore maintain minimum-cost path information in the presence of temporal variation in link quality.

For the introduction of our service, we focus on maintaining specific information about the minimum-cost path, namely, the cost of the path from the node to a single sink, and the neighbouring node on the path closer to the sink, i.e., a preferred parent. Generalizations illustrating the broader applicability of our service are discussed in Section 3.4.

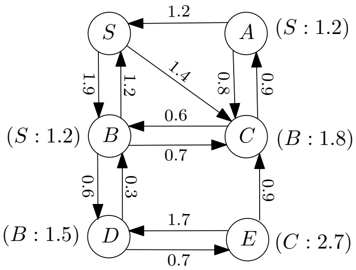


Figure 1: Information about the minimum-cost path to sink S

3.1 Introductory Example

Figure 1 gives an example of a six-node WSN deployment with designated sink node S. The connectivity of a WSN can be described by a weighted directed graph. The vertices of the graph represent the sensor nodes. The directed edges of the graph specify the directed communication links between nodes and are labelled with the cost associated with using them. The cost is typically related to the quality of the link. Many different metrics exist to measure the quality of individual links, typically referred to as Link Quality Estimators (LQEs) [1], based on sent or received packets. Well known examples are the Received Signal Strength Indicator (RSSI), Signal-to-Noise Ratio (SNR) and Packet Reception Ratio (PRR). Our service is independent of the used LQE and therefore, the choice of LQE for a particular deployment is outside the scope of this paper.

We call the neighbouring nodes to which a node has a directed edge in the connectivity graph, and thus can potentially send packets to, the *outbound neighbours*. These outbound neighbours are independent of the used routing algorithm. In the example of Figure 1, the outbound neighbours of node A are nodes S and C. The nodes from which a node can receive messages are called the *inbound neighbours*. In Figure 1, the only inbound neighbour of node A is node C.

A path between two nodes is a sequence of k nodes n_0 to n_{k-1}, such that n_{i+1} is an outbound neighbour of n_i for all $0 \leq i < k-1$. The cost of a path is a function of the cost associated with the communication links on that path. The minimum-cost path of a node is selected from all possible paths from that node to the sink. Given any monotone recursive function f, expressing path costs, the minimum path-cost of a node can be computed using the following recursive procedure, where the sink has a minimum-cost equal to the identity element of the function, i.e., 0 for summation, 1 for multiplication, etc. *lqe(n,x)* equals the cost of the link from node n to its outbound neighbour x.

$$cost(n) = \begin{cases} identity(f) & \text{if} \quad n = Sink \\ \min\limits_{x \in outbounds} f(lqe(n,x), cost(x)) & \text{if} \quad n \neq Sink \end{cases}$$

We can conclude that, given this recursive procedure, nodes can determine the minimum-cost and the parent on that path in a distributed manner using only knowledge of (i) the minimum-cost for all of its outbound neighbours and (ii) the cost of communicating to those neighbours based on LQE information.

For the example of Figure 1, we assume the cost of a link to represent the expected number of retransmissions. Given that we are interested in minimizing the number of retransmissions, the recursive function f is summation (with identity element 0). The non-sink nodes have been labelled with the resulting cost of the minimum-cost path to S and with the parent on that path. For example, node E has two outbound neighbours. The minimum cost of the path using node C as a parent is $1.8 + 0.9 = 2.7$ while using node D results in a cost of $1.5 + 1.7 = 3.2$. The minimum cost can be easily determined to be 2.7 and the parent on the minimum-cost path to be C.

3.2 Communicating Across Asymmetric Links

With symmetric links we can assume that the inbound neighbours of a node are always at one hop distance. Because of this and the fact that the link quality of both the inbound and outbound link are considered to be equal, we can furthermore assume that the cost of communicating to an outbound neighbour can be derived by every node locally. These assumptions make the currently used distributed approach using only a local broadcast of the known minimum cost sufficient to provide nodes with the required information on the minimum cost for and cost of communicating to all outbound neighbours in an iterative fashion.

Asymmetric links violate both assumptions. First of all, communicating information to inbound neighbours may require more than one hop, in case of unidirectional links, and communicating over multiple hops can also have a lower cost (for instance higher probability of successful transmission), in the presence of shorter paths with a bad quality. Secondly, although LQE's based on transmitted packets may still allow nodes to determine the cost of communicating to outbound neighbours locally, the LQE's based on received packets (most of the well-known LQE's) do not. To allow nodes to derive their minimum-cost path, nodes with a known minimum-cost path should therefore communicate both its minimum-path cost and the observed link cost of the inbound neighbours to all inbound neighbours.

In general, the route to take to communicate with an inbound neighbour over multiple hops may be unknown to the sender. Therefore, our service depends on a generic controlled n-hop forwarding approach. Nodes broadcast their information and nodes receiving it directly rebroadcast the received information for a limited number of hops. Redundant forwarding of packets is prevented by not forwarding the same information more than once. By using more than one hop, information can be relayed around unidirectional links (as long as an alternative path exists).

Additional administrative information needs to be added to the packets to allow nodes to determine whether received packets are from one of their outbound neighbours and should be forwarded to inform more nodes. For the former, we add the sender's unique identifier and the list of intended recipients, the inbound neighbours' unique identifiers, to the packet. To determine whether a packet should be forwarded, we add 'hops-to-live' information and a sequence number to the packet. The hops-to-live value indicates the remaining number of hops the information should be forwarded. With the sequence number we can identify the forwarded information. Figure 2 shows the resulting format of the packet sent by nodes with a known minimum-cost path.

Figure 3 shows the benefit of using n-hop forwarding for the distributed derivation of minimum-cost path information compared to the approach of local broadcasting in the presence of asymmetric links. Figure 3(a) shows the result of the

Sender ID	*Path-cost*	*Set of inbound neighbours* *ID + link-cost*	*Seq nr*	*htl*

Figure 2: Format minimum-cost information packet

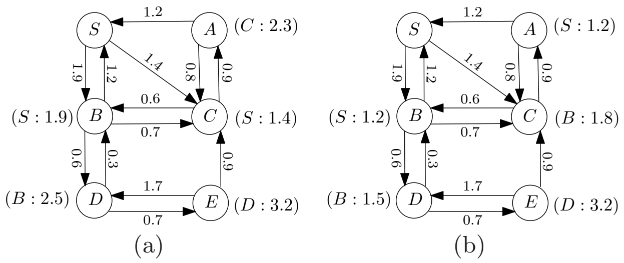


Figure 3: Information about minimum-cost path (a) using local broadcasting (assuming link symmetry) (b) using asymmetric-aware 2-hop forwarding

minimum cost and parent calculation using local broadcasting. Every node with a known minimum cost broadcasts it and nodes receiving this information (inaccurately) assume they can communicate with the node from which they have received the information at the cost determined from the incoming link. Figure 3(b) shows the result of using a 2-hop forwarding approach for every node in the network. While for the local broadcasting approach the cost and parent are inaccurate for almost all nodes (compare Figure 1), the accuracy has been significantly increased. This is due to two properties of the n-hop forwarding approach. Firstly, the cost of outgoing and incoming links is explicitly distinguished (resulting in an increased accuracy for, for example, node B) and, secondly, nodes are informed over more than one hop (resulting in an increased accuracy for node A). Note that for this example a 3-hop forwarding approach would result in an even more accurate solution, which is the optimal solution, as shown in Figure 1.

The example also shows that controlled n-hop forwarding allows for a trade-off between the accuracy of the derived minimum-cost path information and the overhead required to derive it. We want to reach as many inbound neighbours as possible to derive accurate information, while avoiding to increase the traffic load in the network potentially impacting the overall network performance. By adjusting the value of n, the n-hop forwarding approach trades off the number of inbound neighbours reached (and thereby the accuracy of the information) for the overhead incurred. Note that when using 1-hop forwarding, the number of packets communicated is the same as with the local broadcast, while the accuracy is improved due to the explicit separation of outgoing and incoming links in the communicated information. This trade-off is discussed in more detail in Section 4.1.

3.3 Maintaining Information

While the controlled n-hop forwarding allows us to provide inbound neighbours with the required information at a given moment in time, the minimum-cost path may change over time due to dynamic changes in the network. It is therefore important for the accuracy of the maintained minimum-cost path information that nodes are informed about both the changes in minimum cost and the cost of the links towards any of the outbound neighbours.

Algorithm 1: Receive minimum-cost path information: (SenderID, Cost, Inbounds(ID, Linkcost), SeqNr, htl)

```
Data:
$myOUT$ = set of ($ID, path\_cost, last\_update$)
$lastSeqNr[]$: sequence number of last forwarded info for every neighbour
$myID$: unique identifier of the node
$TIME$: current (local) time of the node
for $in \in Inbounds$ do
    if $myID = in.ID$ then
        Add or update ($SenderID, Cost + in.Linkcost, TIME$) in $myOUT$
    end
end
if $htl > 0 \wedge MyID \neq SenderID \wedge lastSeqNr[SenderID] < SeqNr$ then
    Broadcast ($SenderID, Cost, Inbounds, SeqNr, htl - 1$)
    $lastSeqNr[SenderID] := SeqNr$
end
```

A straightforward approach is to send the minimum cost and the inbound link costs using the n-hop forwarding approach every time a change in any of them is observed. This approach can lead to an uncontrollable amount of packets in the network, as there is typically no upper bound on the number of changes. Furthermore, a problem arises with inbound neighbours that are lost over time, for example because they moved out of range. Nodes should become aware of the fact that they lost an outbound neighbour to prevent them from using the stale information. This is hard as the node itself may not be able to observe this change and the outbound neighbour that does observe this can be out of range. To control the overhead of maintenance and to reduce the likelihood that nodes derive their information based on outdated information, we use a flexible approach in which we do not assume information to be valid forever, without reconfirmation, but where received information from an outbound neighbour has a limited *validity interval*.

To provide inbound neighbours with up-to-date information, nodes at a given interval, referred to as the *update interval*, use the controlled n-hop forwarding approach to communicate the packet of Figure 2 if a minimum-cost path to the sink is known. This way nodes receiving the information can maintain up-to-date knowledge about the minimum cost for all of their outbound neighbours and the cost of communicating to those neighbours. Note that it is not necessary to synchronize the update intervals between the nodes. Setting the update intervals allows us to trade off the responsiveness to changes in the network, i.e., the temporal accuracy of the information, with the maintenance overhead. Appropriate setting of the update interval is done in conjunction with the validity interval and depends on the characteristics of the deployment. This is discussed in more detail in Section 4.1.

After receiving minimum-cost path information packets, nodes should process them, check whether it contains new information from its outbound neighbours and forward the information if needed. The pseudo-code of the required actions after receiving the information from a neighbouring node is shown in Algorithm 1.

Now nodes are able to locally maintain the path cost of using any of its outbound neighbours and should regularly check this information for necessary updates. Assuming that the processor time is not a bottleneck and is not a significant part in the overhead, nodes can adapt their minimum-cost information immediately based on the latest updates

Algorithm 2: Update minimum-cost path information

```
Data:
$myOUT$ = set of ($ID$, $path\_cost$, $last\_update$)
$pathKnown$: true if minimum-cost path known by node
$myCost$: maintained cost of minimum-cost path
$myParent$: maintained parent on minimum-cost path
$vi$: validity interval
$TIME$: current (local) time of the node
$pathKnown :=$ false; $myCost := \infty$;
for $out \in myOUT$ do
    if
    $out.path\_cost < myCost \wedge out.last\_update > (TIME - vi)$
    then
        $myCost := out.path\_cost$
        $myParent := out.ID$
        $pathKnown :=$ true
    end
end
```

received. By checking at a regular interval, which we refer to as the *check interval*, we can make an additional trade off between accuracy and used processor time. The pseudocode for updating the minimum-cost information is shown in Algorithm 2.

3.4 Generalization

The service introduced in this paper is primarily intended to compute minimum cost values and paths. The service can however be generalized on several levels, using different link quality metrics or using different definitions of path cost.

In the previous subsections, we have focused our discussion on LQE's which determine quality by *received* packets. For LQE's which are only based on transmitted packets, such as the Required Number of Packet transmissions (RNP) [3], nodes can locally determine the cost to communicate to their outbound neighbours. The controlled n-hop forwarding approach is still required to overcome asymmetric links and communicate the node's cost value, but nodes do not need to communicate their inbound neighbour's ID and link cost, but can use locally determined information instead.

The service can furthermore easily be extended to work with path costs to multiple reference nodes, by adding the minimum cost to each of the reference nodes to the packet. Note that other information, such as the link cost of the inbound neighbours, remains the same, which makes this service scale efficiently to multiple reference nodes. Additional information about the minimal path can be added, such as the complete set of nodes (node IDs) on the minimum-cost path, which may be of interest to routing protocols. In that case, every node should include the unique identifiers of all the nodes on its own minimum-cost path to the communication to its inbound neighbours.

Besides its generality for the problem of maintaining minimum-cost path information, we can further extend the applicability of our service to distributed maintenance of other global properties that fit the pattern of recursive distributed calculation. One could think of the local reachability of services provided by other nodes in the network or deriving information on connectivity to other nodes or clusters in the network.

4. PERFORMANCE EVALUATION

We start this section by discussing the influence of the deployment on setting the service parameters influencing the trade-off between the accuracy and overhead, based on experience with extensive simulations and actual deployments. In Section 4.2, the performance of the service is evaluated for a typical WSN deployment and compared to the local broadcasting approach with the use of extensive simulations. With the design of our service we kept in mind an easy integration into existing and new WSN deployments and, in Section 4.3, we discuss the integration of our service in an actual deployment. While this actual deployment makes it harder to compare two approaches, as interference and mobility patterns are not equal for experiments at different moments in time, it gives us insight in the practical applicability of our service.

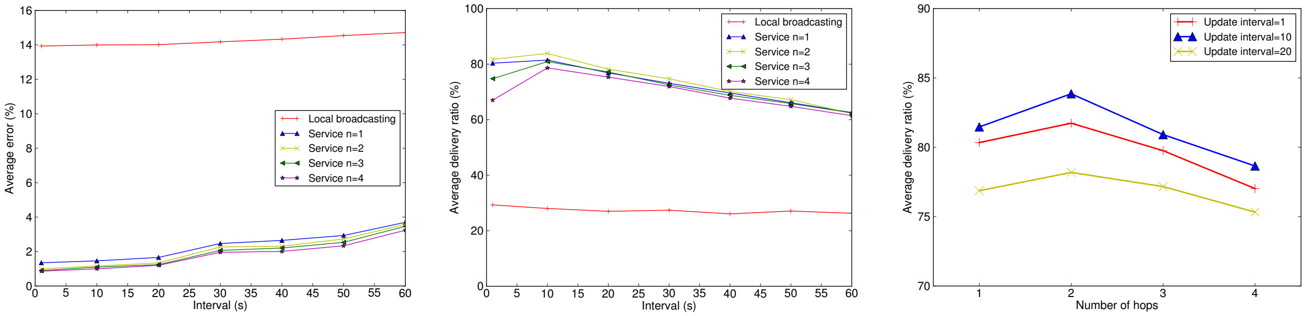
4.1 Setting the Service Parameters

As discussed in the previous section, our service has four parameters making it flexible in integrating it into a given WSN deployment. Namely, the number n, of hops to forward, the *validity interval* of received minimum-cost path information, the *update interval* at which inbound neighbours are informed and the *check interval* at which the minimum-cost path info is updated locally.

For the number n of hops to forward, both the connectivity structure and amount of asymmetry play an important role in how much hops it takes to communicate to inbound neighbours. At design-time we can get a good idea about the connectivity structure and the amount of hops needed to communicate to inbound neighbours by looking at the (worst-case) difference between transmission range and the potential locations of nodes. In [5] it is observed that forwarding the information over two or three hops is enough to overcome the asymmetry in typical deployments. Also for the performed simulations and experiments as discussed later we observe that a number of hops larger than two does not significantly increase the accuracy of the maintained minimum-cost path information.

The *validity interval* should be set in conjunction with the *update interval* and should be selected to be at most the validity interval to avoid that nodes frequently assume to have no valid information. To take packet loss into consideration within the service, we can furthermore set the update interval such that the nodes are updated multiple times within the validity interval. The update interval parameter is found to be the main driver for the trade-off between accuracy and overhead of the service. Setting the value of the parameter is guided by the amount of dynamic changes of the link quality. While external interference impacts the link quality over time, we see from experience, that the speed of mobile nodes is the most important source of dynamic changes to the link quality. Currently, our service is found to be very well applicable for the speed of mobility that is typically found in the monitoring of persons or animals (see also the next section), which make up most of the current WSN deployments. Responding to very high dynamics requires a very short update interval to provide accurate minimum-cost path information, resulting in a large overhead of our service or any other approach to maintain global knowledge locally. The practical use of global information, such as the minimum-cost path, in such highly dynamic networks is questionable, and currently not the focus of our service.

In the previous section, we reasoned that the *check interval* can usually be chosen sufficiently low or we can even check the maintained minimum-cost path information of out-



(a) (b) (c)

Figure 4: Impact on average (a) error (b) delivery ratio. (c) Average delivery ratio vs. number of hops

bound neighbours every time it changes. This is also observed in practice as the amount of changes to the information is controlled and the check itself requires only a small amount of processor time.

The accuracy of the LQE is found to be an important and non-trivial aspect in the complete solution of maintaining minimum-cost path information. Future work will also focus on a more detailed exploration of the relation between the chosen LQE and the performance of our service.

Finally, it is important to note that the amount of overhead as introduced by our service compared to the local broadcasting approach is inevitable when a local broadcasting approach does simply not suffice to provide sufficiently accurate information.

4.2 Simulation Results

We implemented both our service and the local broadcasting approach as a separate module in OMNeT++ [13], a discrete-event simulation environment, with the use of MiXiM [11], a modeling framework created for wireless networks. For the simulations with the service, we fixed the validity interval to be two times the update interval and the check interval to be one second. For the model of the local broadcasting approach we furthermore made a simple extension which allows to repeat the broadcasting at a certain interval as a single broadcast is clearly unable to handle minimum-cost path changes over time.

The nodes used in the simulated WSN are modeled to have the characteristics of the popular TelosB nodes [18]. The MAC protocol used by the nodes is the non-beacon IEEE 802.15.4 protocol without acknowledgements, as provided in the MiXiM framework. We furthermore implemented a simple minimum-cost routing protocol, which uses the parent on the minimum-cost path for forwarding application packets. Assuming link transmissions are independent, the probability of successful delivery, i.e cost of the path, can easily be calculated by the product of the probabilities of successful transmission of all links on the path. For the simulations we require an accurate estimator for the cost of an individual link. The average observed Bit-Error-Rate (BER) received in the last number of seconds equal to the validity interval is found to result in a good estimation of the cost for the used scenario model. The BER is a function of the receiver Signal-to-Noise Ratio (SNR), which is a hardware LQE integrated in the TelosB CC2420 chip-set.

For our experiments we look at a typical WSN scenario in which several mobile nodes move through an area covered with static nodes. We look at an average sized network, with a hundred static nodes positioned in a grid-like fashion with ten meters between them. There are ten mobile nodes that repeatedly move at walking speed (2 m/s) to a random location in the 100 by 100 meter area and stay at this location for a random amount of time, complying to the Random Waypoint Mobility model [7]. At an interval of one second the mobile nodes send a single packet to the sink, which is located at the left upper side of the static grid of nodes. The transmission power of the static nodes is selected to be -5 dBm, which represents a transmission range of around 12 meter for the used model, allowing the nodes in the grid to be connected to their direct neighbours. The mobile nodes have a lower transmission power of -15 dBm with a range of around 7 meter, which is enough to reach at least one node in the static grid. The fact that static and mobile nodes have different transmission powers results in communication links with asymmetric or even unidirectional quality. Due to the mobility of the nodes, the link qualities and connectivity change over time, resulting in time-varying minimum-cost paths.

For a single experiment, every mobile node sends 10,000 messages to the sink. The nodes start sending packets after an initialization phase, in which nodes initialize their minimum-cost path information. The measured experimental data is collected after the initialization until the nodes have sent all their messages to the sink. To have statistically more reliable results, every experiment was repeated 10 times and shown results are the average over all runs.

By simulating the above scenario, we can observe the cost of the minimum-cost path derived with both the local broadcasting approach and the service and compare it with a simulated oracle that computes the actual minimum cost. With this knowledge we can derive an error for every node by averaging the absolute error between the derived and actual minimum cost over time. Figure 4(a) shows the average of this error for the ten mobile nodes for different intervals at which the broadcasting is repeated and the nodes inform neighbouring nodes, i.e., the update interval. We furthermore vary the number of hops used for n-hop forwarding.

We can make several observations based on these simulation results. First, the error of the local broadcasting approach is significantly larger than for the service, independent of the chosen interval and number of hops. This shows that for the accuracy of the cost derivation the importance of accounting for asymmetric links is large. We also see that reducing the interval at which information is

Table 1: Packets per broadcasting/update interval

Local broadcasting	*Service* $n = 1$	$n = 2$	$n = 3$	$n = 4$
1	1	4.5	11.8	18.2

distributed has a positive effect on the error as the maximum time during which wrong information might be used is reduced. For the chosen scenario, increasing the number of hops for n-hop forwarding increases the accuracy, but the effect is found to be much less than the increased accuracy of only taking asymmetry into account.

When the accuracy of the cost derivation is increased, it is expected that the delivery ratio will increase as the routing parent is more accurately chosen. In Figure 4(b), we can see the results of the average delivery ratio of the ten mobile nodes while varying the parameters of the derivation approaches. As expected, we see that the service results in a much larger delivery ratio, independent of the chosen parameters. Several interesting observations can be made. While the accuracy is invariably increased by reducing the update interval and increasing the number of hops, we see that the delivery ratio shows an optimum parameter value, below which the delivery ratio decreases again. For this deployment, reducing the update interval to lower than 10 seconds reduces the delivery ratio even though the error is reduced. The same holds for increasing the number of hops. The delivery ratio has a maximum when the number of hops is 2, above which the delivery ratio is reduced. These effects can be observed better in Figure 4(c), where we plot the average delivery ratio versus the number of hops for the ranges of interest. Both effects can be explained by the fact that the interference caused by the increased overhead of the service outweighs the benefit of the (here small) increase in accuracy for the simulated scenario.

In Table 1, we show the average number of packets per broadcasting/update interval for the given scenario. From these numbers, we see that the number of packets is increasing with the number of hops, due to an increased number of neighbours for which information needs to be forwarded.

What we have observed from these simulations is that the service can significantly increase the accuracy of the maintained minimum-cost path and the efficiency of the protocol relying on this information at the expense of an amount of overhead. Using the local broadcasting approach without repeated broadcasting is clearly no realistic approach in a dynamic network and even by repeating the broadcasting, accuracy is low and the routing protocol is not able to deliver more than 30% of the packets. We furthermore observed that at some point of the parameter settings, the accuracy of the service may still increase, but the induced overhead reduces the efficiency of the protocol. Depending on the delivery ratio required by the application, the overhead can easily be reduced by lowering the update interval or number of hops to forward at the cost of only a slight reduction in delivery ratio.

4.3 Experiments With an Actual Deployment

We implemented our service and the repetitive local broadcasting approach in TinyOS 2.x [19]. We experimented with these two approaches for a heterogeneous WSN deployment in our office building. We use a similar scenario as used for the simulations. Four employees are equipped with a BSN node [2], which has a similar design as a TelosB node but has a smaller form-factor and integrated accelerometer, making them ideal as mobile node. These mobile nodes communicate packets to a sink PC at a rate of 1 every 5 seconds using a static network consisting of 15 TelosB nodes covering the areas of interest for the four monitored employees.

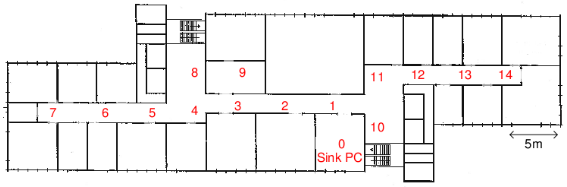


Figure 5: Experimental deployment of the 15 static TelosB nodes, monitoring 4 mobile BSN nodes

We implemented and used the same protocol stack as used in the simulations, i.e., an IEEE 802.15.4 MAC and a simple minimum-cost routing protocol. We fixed the validity interval of outbound neighbours to be two times the update interval and nodes adapt their minimum-cost information immediately based on the latest updates received. After several small-scale experiments we found the average RSSI value of the packets received in the last interval of time equal to the validity interval to be a good indication of the link quality [16].

We selected the transmission power of the nodes to be -15 dBm, which allows a reasonable distance between the nodes while it is still low in order to preserve energy spent by the radio. As the antenna characteristics, e.g., size, sensitivity, differ between the BSN and TelosB nodes, the maximum transmission range of a node depends on the type of node involved in the communication. The observed maximum transmission range from a TelosB to TelosB node is around 13 meter while being 7 meter from TelosB to BSN. The maximum range of a BSN node communicating to a TelosB node is observed to be around just 3 meter.

The TelosB nodes are deployed such that the distance between a mobile and static node is at most 2.5 meter in the areas where we monitor the employees, allowing every mobile node to be able to directly communicate to the static network. The resulting deployment is shown in Figure 5.

For a single experiment we let the mobile nodes send packets to the sink for half an hour after an initialization phase allowing all nodes to initially derive their parent. During the experiment the mobile nodes move through the building based on typical daily activities of the employees, i.e., work in an office, gather at the meeting areas and walk between these areas. After every experiment every node sends packets with statistics information for performance analysis on the sink PC.

Figure 6 shows the resulting average, as well as the minimum and maximum, delivery ratio observed in our experiments with the two minimum-cost path derivation approaches while varying the broadcasting/update interval. The number of hops used by the service for the shown results is 1. For experiments done with a higher number of hops we have seen no significant improvement and even observed slight reduction of delivery ratio. This is because, for the asymmetry as observed in this deployment, we see that one hop is typically sufficient to receive a packet from the

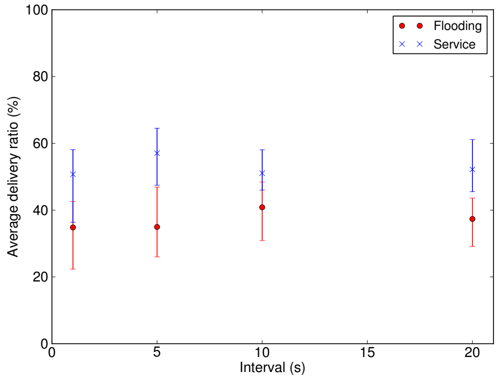


Figure 6: Impact of two approaches on delivery ratio

parent on the minimum-cost path, allowing multiple hop forwarding only to give a very limited improvement (similarly as observed in the simulations).

First of all, for both approaches the delivery ratio is fairly low. This mainly has to do with the chosen protocol stack. We have chosen to focus on a simple protocol stack, but this approach lacks robustness to packet loss as only a single path to the sink is considered and no acknowledgment approach is used (as this should also be adapted to deal with asymmetric links). Even though the delivery ratios are low, for the current experiments we clearly see an improvement in the average delivery ratio, up to 22%, when using our service compared to the local broadcast approach.

Between experiments it was hard to replicate exact mobility patterns and the external interference can not be controlled at all. Currently, the impact on the delivery ratio of these variations between experiments can not be quantified. We therefore can not give conclusive statements on the optimal parameter values with the current experiments. Future work will focus on more extensive experiments with larger networks exploring various protocol stacks.

The experiments with the actual deployment gave us feedback on the practical applicability of the service and allowed us to get more insight in the issues and trade-offs involved as also described at the beginning of this section. The service is found to allow an easy integration into protocol stacks for a practical WSN deployment. From the current experiments we did observe a significant positive impact on the delivery ratio when using the service compared to the typically used local broadcasting approach.

5. CONCLUSIONS

In this paper, we have introduced a distributed service that allows nodes to maintain accurate information about the minimum-cost path for deployments with links with an asymmetric and varying link quality. The service requires nodes to use controlled n-hop forwarding to communicate the required minimum-cost path and connectivity information across asymmetric links. To maintain accuracy after dynamic changes in the link quality, the controlled n-hop forwarding is repeated at a given interval to update nodes depending on this information.

The design and flexibility of our service allow it to be easily integrated into existing and new WSN deployments. We provided insights on the characteristics of a deployment that impact the accuracy of derived minimum-cost path information and the overhead to maintain it. We furthermore discussed how to set the parameters of the service accordingly. With the use of extensive simulations and experiments with an actual deployment, we explored the impact of the parameters on the accuracy and overhead of the service for a given deployment. Simulations and experiments furthermore show a significant increase in accuracy of the maintained information and efficiency of the protocol using this information compared to the typically used local broadcasting approach.

6. REFERENCES

[1] N. Baccour et al. A comparative simulation study of link quality estimators in wireless sensor networks. *MASCOTS*, pages 1–10, 2009.

[2] BSN website, ICL London. http://vip.doc.ic.ac.uk/bsn/m621.html.

[3] A. Cerpa et al. Temporal properties of low power wireless links: modeling and implications on multi-hop routing. *MobiHoc*, pages 414–425, 2005.

[4] J.-H. Chang and L. Tassiulas. Maximum lifetime routing in wireless sensor networks. *IEEE/ACM Trans. Netw.*, 12(4):609–619, 2004.

[5] J. Du, W. Shi, and K. Sha. Asymmetry-aware link layer services in wireless sensor networks. *Journal of Embedded Computing*, 3:141–154, 2009.

[6] R. Fonseca et al. Tinyos tep 123: The collection tree protocol. August 2006.

[7] E. Hyytiä and J. Virtamo. Random waypoint mobility model in cellular networks. *Wirel. Netw.*, 13:177–188, 2007.

[8] C. Intanagonwiwat et. al. Directed diffusion for wireless sensor networking. *IEEE/ACM Trans. Netw.*, 11(1):2–16, 2003.

[9] D. Kotz, C. Newport, and C. Elliott. The mistaken axioms of wireless network research. Technical Report TR2003-467, Dartmouth College, July 2003.

[10] S. Lin et al. ATPC: adaptive transmission power control for wireless sensor networks. In *SenSys*, pages 223–236, 2006.

[11] MiXiM website. mixim.sourceforge.net.

[12] D. Niculescu and B. Nath. Ad hoc positioning system (APS). In *GLOBECOM*, volume 5, pages 2926–2931, 2001.

[13] OMNeT++ website. www.omnetpp.org.

[14] L. Sang, A. Arora, and H. Zhang. On link asymmetry and one-way estimation in wireless sensor networks. *ACM Trans. Sen. Netw.*, 6(2), 2010.

[15] C. Schurgers and M. Srivastava. Energy efficient routing in wireless sensor networks. *MILCOM*, 2001.

[16] K. Srinivasan and P. Levis. RSSI is Under Appreciated. In *EmNets*, 2006.

[17] I. Stojmenovic and X. Lin. Power-aware localized routing in wireless networks. *IEEE Trans. Parallel Distrib. Syst.*, 12(11):1122–1133, 2001.

[18] TelosB Datasheet, Crossbow Inc. www.xbow.com.

[19] TinyOS website. www.tinyos.net.

[20] A. Woo, T. Tong, and D. Culler. Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks. In *SenSys*, pages 14–27, 2003.

[21] F. Ye and A. Chen. A Scalable Solution to Minimum Cost Forwarding in Large Sensor Networks. In *ICCCN*, pages 304–309, 2001.

[22] F. Ye et al. GRAdient broadcast: a robust data delivery protocol for large scale sensor networks. *Wirel. Netw.*, 11(3):285–298, 2005.

[23] J. Zhao and R. Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *SenSys*, pages 1–13, 2003.

[24] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic. Impact of Radio Irregularity on Wireless Sensor Networks. In *MobiSYS*, pages 125–138, 2004.