Practical and self-configurable multihop wireless infrastructure: a functional perspective

Muhammad Hamad Alizai

Department of Computer Science,
LUMS SBASSE,
54792 Lahore, Pakistan
Email: hamad.alizai@lums.edu.pk

Abstract: Multihop wireless networks require self-configurable and reliable communication infrastructure that can quickly adapt itself to the rapidly changing network conditions without manual reconfiguration. The key to developing such a communication infrastructure requires (i) efficient link estimation mechanisms, (ii) reliable routing algorithms, and (iii) stable addressing schemes. In the past decade, a number of practical solutions have been developed for these three main components of multihop wireless infrastructure in inherently similar mesh networks, sensor networks, and MANETs. However, we lack a comprehensive review that classifies these solutions and compares them on the basis of a set of properties desired in wireless networks. In this review, we classify different techniques implemented and evaluated by the research community, define a set of properties, and functionally compare representative techniques from each class.

Keywords: link estimation; routing; addressing; wireless protocols; sensor networks; mesh networks; multihop wireless.

Reference to this paper should be made as follows: Alzai, M.H. (xxxx) 'Practical and self-configurable multihop wireless infrastructure: a functional perspective', *Int. J. Ad Hoc and Ubiquitous Computing*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Muhammad Hamad Alizai is an Assistant Professor at the Department of Computer Science, LUMS. Previously, he was a Research Assistant at ComSys Group, RWTH Aachen University, Germany. He received his PhD and MSc from RWTH Aachen University in 2012 and 2007, respectively. His research interests are in mobile applications, internet of things, sensornets, and delay tolerant networking. He is particularly interested in applications, protocols, architectures, and evaluation tools for future networks.

1 Introduction

Multi-hop wireless networks, such as sensor networks, mesh networks, and MANETs suffer from permanent topological dynamics due to unstable wireless links and node churn and mobility. Previous studies have shown that wireless links are highly unstable and vary significantly in their quality (Aguayo et al., 2004; Cerpa et al., 2005a, 2005b). Several factors, such as physical distance between a node and its neighbour (Zhao and Govindan, 2003; Becher et al., 2008), environmental conditions (Rappaport, 2001; Puccinelli and Haenggi, 2006), the interference experienced by each link from nearby networks operating in the same frequency range (Huang and Park, 2009; Kim et al., 2009; Qiu et al., 2007; Niculescu, 2007), contribute to these variations among link qualities across a wireless network. Similarly, node churn and mobility is a prominent, complementary factor that prompts unpredictable topological changes. These node dynamics further complicate the provision and maintenance of a reliable network topology in wireless networks. Consequently, unlike wired networks, where the nodes are statically configured for routing and addressing over stable links, multihop wireless networks demand self-configurable[1] and adaptable mechanisms at the network layer to deal with such challenging network conditions.

Achieving multihop communication in a self-configurable wireless network thus deals with three different mechanisms:

- link estimation
- routing
- addressing.

Link estimation is concerned with identifying high quality links within a node's one-hop neighbourhood. These links are typically identified based on the their success rate observed over a certain time frame. Routing protocols use link estimators to establish routing paths in the network that span multiple hops. A straightforward mechanism is to establish a tree-like topology, either proactively or reactively, by selecting the best quality link at each hop that minimises the remaining distance to the destination. Finally, a self-configurable addressing scheme is required to achieve point-to-point communication. A common scheme is to assign virtual coordinates to nodes: Construct multiple trees rooted at landmarks – designated nodes – and determine a node's location based on the vector of hop counts from a set of landmarks.

Another observation is that, due to the structural and topological similarities between different types of wireless networks (e.g., sensor networks and mesh networks), protocols developed for these three mechanisms at the network layer are usually applicable across multiple network types (Landsiedel et al., 2012; Alizai et al., 2013). This is also the reason that the research in different types of multihop wireless networks borrows heavily from each other when it comes to developing scalable, self-configurable, and reliable multihop wireless infrastructure. Expected transmission count (ETX) metric (De Couto et al., 2005), 4BLE (Fonseca et al., 2007), and AODV (Perkins et al., 1999, 2003) are among the examples of such widely used protocols that have simultaneously been used in sensor networks, mesh networks and MANETs implementations. It means that, in most cases, transition from one type of wireless networks to other usually demands requisite adaptations in operational parameters and complexity of the underlying protocols while the core mechanisms still remain the same (Landsiedel et al., 2012; Alizai et al., 2013).

In this paper we revisit the fundamental concepts of link estimation, routing, and addressing in self-configurable multihop wireless networks. We present some of the most prominent case studies that represent the state-of-the-art in these three characteristically similar network types. Although the discussion in this paper covers a broad spectrum of multihop wireless research, the case studies will pay special attention to sensor networks for two reasons. First, when comparing state-of-the-art implementation, we need to maintain a fair comparison base. Second, among others, sensor networks represent notoriously the most difficult and deprived type of multihop wireless networks in terms or resource availability and communication bandwidth.

We first examine the details of each case study at a requisite level to include the pivotal concepts in this paper. However, the core of this paper deals with comparing these studies. In this regard, we define some key properties for link estimation, routing and addressing and rate the case studies on the basis of these properties. Our discussion thus targets the design philosophy of these protocols and not just their performance. For example, we are interested in the scalability and reliability of a protocol design and not the achieved throughput of a particular implementation. Please note that our rating for different protocol properties is comparative and simply enables better understanding of the design tradeoffs among different approaches. This rating shall not be considered as a formal classification of the approaches discussed here. Overall, we believe that the discussion in this paper forms the proper conceptual bases and facilitates a smooth sailing into the state-of-the-art, multihop wireless research.

The remainder of this paper is structured as follows. In Section 2, we discuss link estimation and some of its prominent approaches. Section 3 discusses routing techniques by putting a special focus on sensor networks. Section 4 presents novel addressing mechanisms for self-maintained wireless networks before we conclude this paper in Section 5.

2 Link estimation

Link estimation is the first step towards building scalable and reliable multihop wireless routing structures. In this section, we discuss the basic concepts and requirements of link estimation. We also define the key properties of a link estimator to compare state-of-the-art studies.

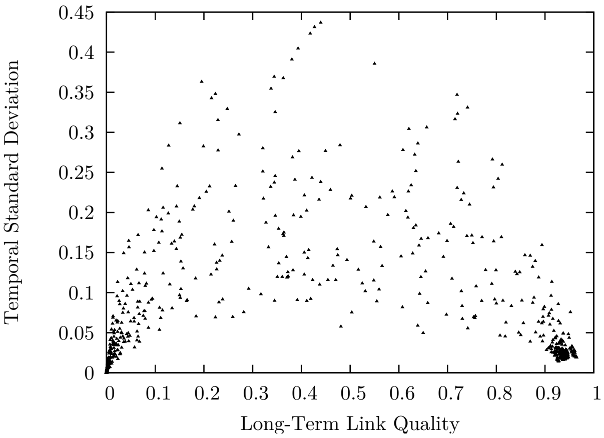
2.1 Introduction

Link estimation deals with identifying high quality links in a wireless network. Depending upon a particular network typed, the term high quality can be used to define a link that optimises throughput, packet loss, congestion, routing progress, energy depletion, or any other form of routing performance measure. However, the predominantly used link metric employed by the majority of today's link estimators (Fonseca et al., 2007; De Couto et al., 2005) is throughput. It is measured in terms of packet reception rates (PRR) or, its reciprocal, ETX: the number of retransmissions required by a packet to reach its destination.

The main challenge in link estimation is that wireless links exhibit strong fluctuations in their quality, especially, when their quality is measured in terms of PRR. For example, Figure 1 shows that for *intermediate links* ($0.1 < PRR < 0.9$), these fluctuations strongly deviate from their $mean$ values. Using such links for data transmission can be detrimental for the performance of a network. Hence, the main task of link estimation is to identify *good links* ($PRR > 0.9$) in the network and to limit packet transmissions to only a selected set of these links.

A *link estimator* estimates the quality of a link from recent transmission traces. The idea is to use transmission traces of sufficient length that minimises the estimation error, i.e., keeps it within $\pm 10\%$ of the actual value. For a link to be scored, it has to be in the *neighbour table* maintained by link estimators. This is because a link estimator only stores transmission traces for links in its table. In order to facilitate scalable network structures, the size of this table is kept constant regardless of the node density. Similarly, other constraints may also apply depending upon the type of wireless network. For example, severe energy constraints in sensor networks strongly limit the computational requirements and the transmission overhead of a link estimator.

Figure 1 Wireless links exhibit inevitable fluctuations in their quality. The long-term link quality represent the PRR for entire experimental run. Each data point represents the standard deviation in PRR calculated over smaller time intervals for each directional node-pair. The graph shows data from an IEEE 802.15.4 based sensor network deployment



2.1.1 *Table management*

Besides the accuracy of link quality estimates, an efficient strategy for neighbour table management is critical in expressing the performance of a link estimator. Table management typically deals with the following three operations.

Link insertion: After receiving a packet from a neighbouring node, the link estimator performs one of the following operations:

- Update the link quality if the link already exists in the table
- insert the link in the table if the table is not already full
- ignore the link
- evict a previous entry from the table and insert this new link.

A link estimator has to carefully choose from these four options ensuring in the meantime that there are enough good links in the table that can be used for data transmissions.

Link reinforcement: This operation deals with reinforcing the quality estimate of a link that already exists in the table. The thresholds for link reinforcement process, such as how often to perform it, has to be selected carefully to ensure that the newly calculated link quality does not overshoot or undershoot the desired accuracy threshold. Here the main tradeoff is between the agility and stability of the link quality estimates. Agility means assigning more weight to the recent estimates for adapting link quality to the most recent underlying link conditions. However, current link estimators prefer stability over agility by tuning parameters that control the history and the weight of the past estimates.

Link eviction: Finally, a link estimator has to determine when to evict a link from the table. Commonly, a time-out or minimum data rate is associated with each link to detect node failures and evict corresponding links. Similarly, a minimum quality threshold is typically defined to evict links whose quality declines below that threshold. An efficient link eviction policy is important to evict unused links and make room for new, potentially valuable links in the table.

2.1.2 *Key properties*

After discussing the basic operational details of link estimators, we now define key properties which, in our opinion, are essential for the design of a link estimator. These properties will help us rate the state-of-the-art techniques and shed light on their benefits and drawbacks.

- *Stability*: This property states the stability of the link estimator both in terms of link estimates and its ability to support a stable routing topology.
- *Adaptability*: It determines how quickly the link quality estimates converge to within the desired accuracy threshold and how well a link estimator adapts its tables to the underlying link dynamics.
- *Current link state*: To see if the link quality reflects the exact link conditions at the time of data transmission or if it is only based on past statistics derived from periodic beacons. This is important because at the time of data transmission, networking conditions, such as traffic patterns and congestion, can be different.
- *Reception correlation*: To determine if packet reception and loss events over a link are considered correlated or independent from each other. This is important because a number of recent studies (Zhu et al., 2010; Srinivasan et al., 2010) have shown that packet loss events on wireless links are temporally and spatially not independent. Hence, incorporating this phenomena in a link estimation implementation is essential to achieve requisite performance benefits.
- *Overhead*: This is the overhead introduced by a link estimator in terms of computational complexity, number of transmissions, and packet overhearing. The transmission overhead includes active link beacons/signalling or additional link estimation information appended with each data packet. Moreover, packet overhearing also introduces significant overhead as a node has to receive and process packets that are not addressed to it.

2.2 *Case studies*

We can divide current link estimation mechanisms into two broad categories, *long-term* and *short-term*.

In long-term link estimation, link qualities are estimated based on the delivery history of a link. We use the term *long-term* to emphasise that the focus of such link estimators lies on the long-term behaviour of a link in the past. In a typical setting, each node snoops the channel for ongoing communication in a network, possibly both for periodic beacon packets and data transmissions. The packet loss over a link is inferred by assigning a unique sequence number to packets from each source. An ETX value is calculated over a window of size t: If n out of N packets are received during t then its ETX is N/n. Commonly, an *exponentially weighted moving average* (EWMA) is used over the past ETX values. Nodes also exchange their link estimates with neighbours to aggregate bidirectional link quality. 4BLE (Fonseca et al., 2007), ETX (De Couto et al., 2005) and BVR's link estimator (Fonseca et al., 2005) are among the prominent derivatives of this method.

On the other hand, short-term link estimation tries to predict the quality of a link based on instantaneous conditions. It does not necessarily maintain any recent history of a link but uses current link state (e.g., by sending active probes) to determine link availability at the time of data transmissions. The supporters of this mechanism argue that the link quality estimates derived from the transmission history of periodic beacon packets do not represent the current state of the link. For example, in sensor networks the network traffic is generated by a rare occurrence of a nondeterministic event. Hence, the channel conditions, such as congestion, experienced by beacon packets in the past are completely different. SOFA (Lee et al., 2006), STLE (Becher et al., 2008), LOF (Zhang et al., 2009) and DUTCHY (Puccinelli and Haenggi, 2008) belong to this category of link estimation.

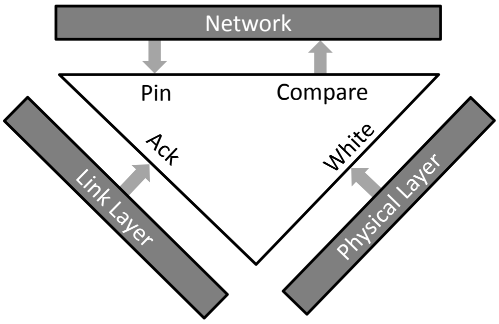
We now present a case study from each of these two categories. Before concluding this section we also present an approach, namely 4C (Liu and Cerpa, 2011), that combines the advantages of both these categories.

2.2.1 Four-bit link estimation

The four bit link estimator (4BLE) (Fonseca et al., 2007) is a state-of-the-art and classical example of a long-term link estimation. It couples link estimation information from broadcast beacons and unicast data transmission resulting into a hybrid ETX for each link. Moreover, 4BLE extends traditional ETX based estimation mechanism by combining information from three different layers – physical, link and network layers – to perform better table management. The key idea behind 4BLE is that each of these layers can provide useful information which benefits link estimation process. For example, the network layer can tell which links are most useful for routing and upper layer applications thereby facilitating a link estimator in link insertion and eviction decisions. Similarly, the physical layer can provide channel quality related information that helps a link estimator in distilling poor links from the estimation process. Overall, 4BLE defines four narrow interfaces to retrieve the following four bits of information – one from physical, one from link, and two from network layers (cf. Figure 2):

- *Pin*: The network layer can pin an entry in the table, preventing the link estimator from evicting this entry. This bit prevents the link estimator from evicting a useful entry from the table.
- *Compare*: It helps resolving inconsistencies between link estimation and routing tables. A link estimator can ask the network layer to compare a newly discovered link with an old entry in the table. The network layer responds by setting the compare bit to suggest that the route provided by the new link is better than the link already occupying the table. This bit helps a link estimator in identifying progressive link from routing perspective and estimate their quality instead of wasting critical resources over a useless link.
- *Ack*: This is the acknowledgement bit set in the transmit buffer if a packet transmission has been acknowledged by the receiver. This bit is used by the estimator to update the corresponding unicast link ETX.
- *White*: This bit reveals the channel quality per packet. A set white bit indicates high channel quality, which means that each bit in the packet has a very low decoding error probability. The white and compare bits are used conjointly to evict entries from the table: If the white bit for a packet received over a newly discovered link is set, the link estimator triggers the procedure corresponding to compare bit in order to decide if this link shall be inserted in the table by removing a random unpinned entry.

Figure 2 The 4BLE uses four bits of information: *Compare* and *Pin* bit from network layer, *Ack* bit from link layer, and *white* bit from physical layer to enhance unicast link estimates and table management policy



Source: Fonseca et al. (2007)

Rating: Figure 3 rates the performance of 4BLE against the properties discussed in Section 2.1.2. The rating scales from one (low) to five points (high). The positive or negative meaning of these scales depends upon the property itself. For example, in the case of scalability, a rating of one point means *poor* scalability, whereas, in the case of overhead, a rating of one point is interpreted as *very good*, indicating small overhead.

By relying on a long-term delivery history of a link of both broadcast beacons and unicast data transmission and extracting useful information from adjacent layers, the 4BLE is by far the most stable current link estimator. CTP (Gnawali et al., 2009, 2013), a widely used collection protocol (cf. Section 3.2.1), uses 4BLE and outperforms contemporary approaches of routing in sensor networks by maintaining a stable and a flawless topology. Therefore, it is assigned five points for stability.

4BLE uses an adaptive beaconing mechanism that increases beacon sending rate if a new node is added to the network or if routing inconsistencies (e.g., loops) are detected. This mechanism allows 4BLE to quickly converge its link estimates for a newly added link within the error threshold bounds. Similarly, 4BLE reacts quickly to link failures, i.e., after five failed data transmissions, by disqualifying the link for routing purposes. However, the adaptability of the 4BLE is only limited to such situations, it fails to quickly recognise the underlying link dynamics to improve performance (Alizai et al., 2009, 2012): For example, if a previously ignored link becomes reliable and offers a significantly better alternative path than the current links in the table, 4BLE is unable to promptly react to such opportunities in the network. This is because there are no data transmissions occurring on this link and the adaptive beaconing slows down exponentially until there are inconsistencies detected in the network. Therefore, it is only assigned two points for adaptivity.

Figure 3 The performance rating and the use case for four bit link estimator

Stability	Adaptability	Current link state	Reception correlation	Overhead	Note
●●●●●	●●○○○	●○○○○	●○○○○	●●●○○	Long-term estimation

The link estimates in 4BLE are based on past delivery traces and does not regard the current state of the link. We still assign it one point because it actively monitors data path using ack bit and includes this information in calculating link estimates. In general, packet success and loss events over a link are considered independent from each other. However, it is assigned one point because it disqualifies a link after just five failed data transmissions. The overhead of 4BLE is moderate because

- it only maintains a subset of neighbouring nodes in the table for link estimation
- uses active beaconing (link probes) to exchange link estimation information with other nodes.

However, it does not employ packet overhearing and therefore is assigned three points.

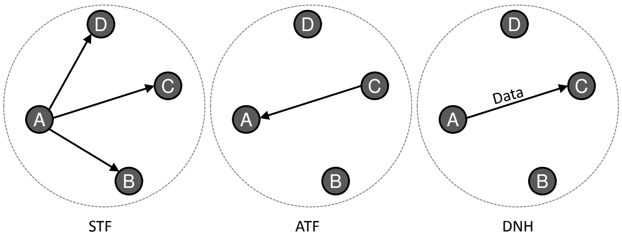
2.2.2 Solicitation based forwarding

Long-term link estimation tries to portray what can be expected from a link in the future based on how it behaved in the past. However, precisely this approach is its major drawback as well. For example, traffic patterns in sensor networks are bursty: The network is in idle state most of the time and only generates large volumes of traffic when a certain event is detected in the environment. Hence, the link estimate derived from its past transmission statistics, i.e., when the network was in idle state, does not accurately reflect the actual quality of the link at the time of transmission. This is because the traffic patterns and congestion in the network are completely different at the time of traffic burst than when idle. Similarly, the active link probes transmitted when the network is in idle state are illusive and consume needless energy.

To address this problem, Lee *et. al.* present Solicitation based forwarding (SOFA) (Lee et al., 2006) that uses a two way handshake to determine link availability. SOFA is not just a link estimator but a complete routing infrastructure for low-power wireless networks. However, its major contribution is the link estimation mechanism. The routing approach of SOFA is based on greedy hop-by-hop forwarding.

SOFA introduces a reactive two-way handshake protocol to determine link availability at the time of transmission. It does not maintain any other information (e.g., quality estimates) regarding a link. Each node, when needing to route a packet to sink, broadcasts a request called *solicit-to-forward* (STF). For example, in Figure 4, node A sends an STF received by its three neighbours B, C and D. A neighbouring node receiving this message can respond with a reply message called *accept-to-forward* (ATF). In our example, C is the first node to reply with an ATF. After receiving this reply message, the sender node makes the replying node its *designated-next-hop* (DNH) and starts forwarding its data as shown in the final step in Figure 4. The DNH is only determined on demand: A timer is associated with DNH and once a node is finished sending its packets for the current event and the timer expires, the node has to redetermine its DNH using the same handshake mechanism. SOFA also employs a passive acknowledgement mechanism: After forwarding a packet, the sender tries to overhear the transmission of its DNH. If it overhears the same packet that was recently forwarded, it implicitly assumes that the packet has been successfully delivered to the DNH. Otherwise, it retransmits the packet until DNH receives it or the maximum number of retransmissions are reached.

Figure 4 The two-way handshake in SOFA. Node A sends an STF message. Node C is the first neighbour to reply with ATF. Node A selects node C as its DNH and forwards data



An important question is how does a node receiving an STF message determine if it is closer to the sink than the sender node. To this end SOFA assigns *height* to each node so that a receiver node can determine its location with respect to the sender node. The idea behind assigning heights is remarkably simple once understood. The sink node sends broadcast advertisements which are disseminated in the whole network. The advertisement is initialised with a height of zero (the sink node has zero height) and incremented by one at each hop as it propagates through the network. Hence, every node knows its relative height from the sink node. A sender always sends its height in STF and a receiver only replies with ATF message if its height is less than the sender's. This mechanism is useful in avoiding routing loops as well. SOFA also employs *height maintenance* mechanism if inconsistencies are observed in the network. For example, if a node's height becomes a local minimum and no other node is replying with an ATF. This shall never happen in a fully connected network.

In dense networks there may exist a large number of neighbours that can reply with ATF. SOFA uses packet overhearing to limit the number of ATF responses. Consider an example with three nodes – A, B and C – each within the communication range of the other. Suppose node A wants to send data and thus broadcasts an STF, which is received by nodes B and C. Lets assume node B replies with an ATF before node C does. In this case, node C will snub its ATF because it has already overheard the ATF of node B. However, this mechanism only works if the neighbouring nodes are within the communication range of each other.

Rating: Figure 5 shows SOFA's rating. As oppose to 4BLE, SOFA neither assigns link estimates nor maintains neighbour tables. Its only contribution towards stability is the height maintenance mechanism which remedies inconsistent topology due to high node churn in the network. Therefore, SOFA receives just one point for stability. The adaptability of SOFA is similar to 4BLE because it only adapts its link selection when *bad conditions*, such as lost transmissions or node failures, occur. However, it does not respond to the opportunities that appear during the course of transmission on other, potentially valuable links. For example, in SOFA, if a neighbouring node offers better routing progress than the current DNH, the sender node will not change its DNH during a transmission burst. Therefore, it gets only two points for adaptability.

Figure 5 The performance rating and use case for SOFA

Stability	Adaptability	Current link state	Reception correlation	Overhead	Note
●○○○○	●●○○○	●●●●●	●●○○○	●●●●○	Short-term estimation

SOFA only uses the current link state information and hence it gets maximum points for this property. Similar to 4BLE, packet loss events are in general considered independent from each other by SOFA. However, the two-way handshake mechanism extrapolates a notion of packet reception correlation since the last packet delivery (i.e., STF packet) is considered sufficient for the success of succeeding transmissions (two points). Although SOFA does not require link tables for its operation, it has a very high communication overhead both in terms of the two-way handshake and the packet overhearing that consumes a significant amount of energy. Especially, the two-way handshake can be detrimental for network performance in challenging networking conditions when a node has to repeatedly select its DNH.

2.2.3 *Foresee (4C) link prediction*

Both long-term and short-term link estimation mechanisms have their own advantages and disadvantages. 4BLE maintains a stable routing topology in the network at the cost of slow-adaptation to underlying link conditions, i.e., by ignoring progressive links that may become reliable during the course of transmission. On the other hand, SOFA uses the current link state for making its decisions, however, many of its design mechanisms are debatable. For example, it has been experimentally demonstrated that a single successful transmission cannot be considered as sufficient evidence of good link conditions (Srinivasan et al., 2008). Moreover, its DNH selection mechanism is very inefficient: In the case of multiple nodes competing to become DNH, a sender node selects a neighbouring node as DNH from which it receives the first ATF response. Hence, it ignores the possibility of using other potentially valuable neighbours.

4C (Liu and Cerpa, 2011) tries to combine the advantages and eliminate the disadvantages of both these techniques. It argues that a stable routing topology is imperative for establishing reliable and robust routing structures. However, a subtle design of a link estimator that explores transmission opportunities over long range intermediate links does not disrupt the stability of today's routing protocols. Hence, it combines short-term link prediction with long-term link estimation to identify short-term high quality links to reduce the number of hops and transmission costs. 4C is a machine learning approach that combines physical layer information with PRR to predict the short-term temporal quality of a link. The prediction models takes an an input the PHY layer parameters and the long-term PRR of a link to compute probability of the next packet being successfully receive. 4C employs a data-driven approach consisting of three steps

- data collection
- offline modelling
- online prediction, i.e., the actual implementation of 4C in sensor networks.

Data collection steps involve collecting link quality data. The size of this data is empirically measured for accurate modelling. 4C records three parameters for each packet; sequence number, RSSI and LQI. Additionally, to compute SNR, the nodes also periodically measures the noise floor level. This data collection is used to experiment different features and compute, for example,

- *W*: the size of link PRR history to make a prediction
- *I*: the periodicity of prediction.

The choice of these two parameters is critically important to determine the feasibility of 4C in a resource constrained environment such as sensor networks.

The choice of prediction model is constrained by the need that it requires a small training data for lightweight online prediction. For this purpose, three different methods namely Naive Bayes classifier, logistic regression and artificial neural networks were used. The empirical evaluation shows that model training only requires to gather data for a few minutes (2–10 min) and the trained model then only needs a single historical packet for predicting the fate of next transmission over that link with high accuracy.

The online prediction model of 4C is implemented in TinyOS and, just as 4BLE, also uses CTP as its routing protocol. Its empirical evaluation on widely used testbeds shows that it can reduce average transmission costs in sensor networks by 30% when compared with 4BLE.

Rating: We now compare 4C with the existing mechanisms by rating it against our established criteria/properties in Figure 6. It allows the existing mechanisms to utilise communication opportunities that might arise over previously ignored class of links (intermediate quality links), using its short-term prediction mechanism, without disrupting the underlying routing topology (Stability = five points). Similarly, 4C is optimistic in its link selection and allows a routing protocol to adapt to the underlying link conditions both in spreading *good news* and *bad news* to the neighbouring nodes. The good news represents a situation where a long range intermediate link becomes temporarily reliable for transmission. 4C utilises such opportunities as early as a single successful transmission. Similarly, the bad news represents a situation when an intermediate link again becomes unreliable for transmission. 4C avoids overshooting an unreliable link by quickly reverting back to a high quality if transmission fails over an intermediate link (Adaptability = four points, Current Link State = five points).

Figure 6 The performance rating and the use case for 4C link predictor

Stability	Adaptability	Current link state	Reception correlation	Overhead	Note
●●●●●	●●●●○	●●●●●	●●●●○	●●●●○	Busty links

The development of 4C also breaks the assumption of independent packet reception events over a link. It predicts the future of a link by combining long-term estimate with a short-term prediction of a link to utilise correlation of packet reception events over a link (Reception Correlation = four points). This information is essential in determining if a link is useful for packet forwarding or not. To make this concept clear let us consider two links, one which rarely transmits a packet successfully and the other which alternates between reliable and unreliable transmission periods, i.e., it is bursty. Approaches such as SOFA cannot differentiate between these two links because they do not employ any mechanism to determine if the previous successful transmission occurred by chance or if this link is bursty. Similarly, it is unlikely that 4BLE will utilise this link because of its poor ETX estimate in the long-term. Finally, 4C it is an extension rather than a replacement of existing long-term link estimation mechanism. Therefore, in addition to the underlying link estimator, similar to 4BLE, it incurs additional overhead of packet overhearing and link prediction calculation (Overhead = four points).

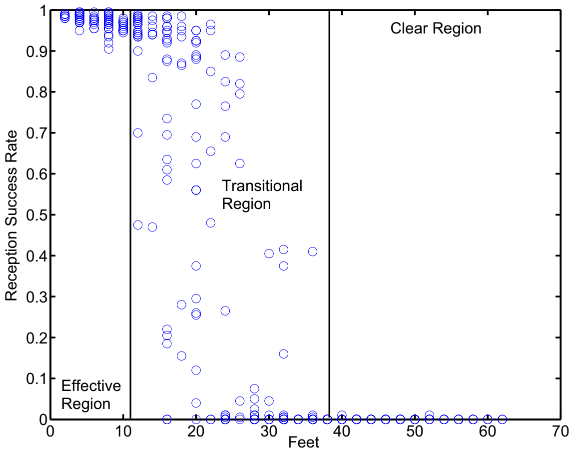
3 Routing

A link estimator is only concerned with a node's one hop neighbourhood. Routing protocols establish multihop structures using link estimation information at each hop. In this section, we discuss some of the prominent routing approaches in sensor networks. We also define key properties of a routing protocol and compare different state-of-the-art approaches.

3.1 Introduction

Unlike wired networks, shortest path routing based on hop-distance metric is not feasible in wireless networks because a wireless link between two nodes reveals more dynamics than simply being considered available or not. For example, a link with $PRR = 40\%$ may deliver enough routing updates to be considered for data packets, thereby resulting in a significant number of retransmissions to deliver a packet. Figure 7 clarifies this observation further by showing the relationship between link reception quality and the distance between communicating nodes. Links from *transitional* and *clear* regions can dominate route selection because they offer better routing progress. However, using these links without assessing their reception quality leads to unstable routing topology, frequent retransmissions, and poor routing throughput.

Figure 7 Reception rates vs. distance between nodes in a line topology: In the effective region all links exhibit good to perfect quality. The quality falls smoothly as the distance between nodes grow (transitional region) and eventually degrading to very poor link quality (clear region) (see online version for colours)



Source: Woo et al. (2003)

In order to deal with these problems, contemporary routing protocols typically employ ETX (De Couto et al., 2005) as a routing metric to establish high throughput paths between distant nodes. Path establishment in multihop wireless networks is usually based on distance vector routing approach: The participating nodes are not aware of the complete network topology. They only know the next hop that leads towards a particular destination and the routing cost along the path offered by that hop. Link state routing mechanisms have also been optimised for multihop wireless settings (e.g., OLSR Clausen and Jacquet (2003), however, they are typically not preferred in large scale settings for two reasons,

- limited scalability
- inherent limitations of wireless devices (especially in sensor networks) in terms of computations, storage and energy.

Routing approaches in wireless networks can be categorised in two broad categories, *address free* and *address based*. In address free routing, a node is typically assigned a unique identifier. It is mostly suited in situations where point to point communication is not relevant such as in data collection and dissemination in sensor networks. Data flows in address free routing can be many-to-one or one-to-many. On the other hand, address based routing is needed for point-to-point communication scenarios where each node in the network can communicate with any other node. Nodes are usually assigned addresses that reveal their topological locations in a network.

We approach these two categories separately: In this section we focus on routing algorithm of a protocol that is used select the next hop for a packet. The next section is devoted to addressing mechanisms that can be used with such routing algorithms.

3.1.1 Tree construction

The majority of self-configuring routing approaches in wireless networks are based on tree construction primitives. Especially, in networks with no access to location services, such as GPS, tree construction is at the helm of establishing scalable routing structures. However, tree construction based routing primitive is not a new concept: It is an established criteria even in wired networks, such as internet back bones, which use the concept of sink trees and spanning trees for each participant in a multicast group (Tanenbaum, 2002).

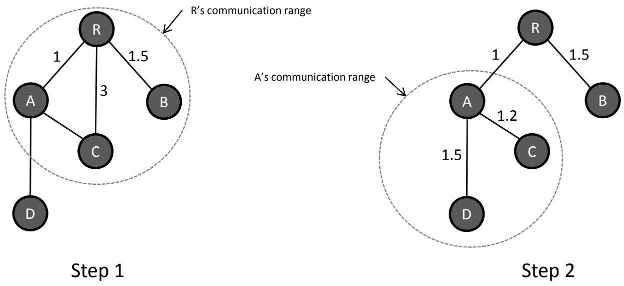
Tree construction resembles the distance vector based routing mechanisms (e.g., Routing Information Protocol (Hedrick, 1988) where each node only maintains its one hop neighbourhood and is unaware of the complete routing graph. For example, if a node X wants to send a packet to a distant node Z, it only knows that it can reach Z through its neighbour Y. However, it has no information, whatsoever, about the nodes on the remainder of the path from Y to Z.

We explain the tree construction phenomena by considering a simple example shown in Figure 8. A tree root R, i.e., a sink in sensor networks or an internet gateway in mesh networks, advertises itself with a distance of 0. The distance can be represented by any metric of interest such as hop count or ETX. In this example, we consider ETX as a routing metric. Each node determines its bidirectional ETX from its neighbours using active link probes as discussed in Section 2. In the first step (cf. Figure 8), nodes A, B, and C receive this advertisement as they are within the radio range of root R. As this direct link is the only choice currently available to reach R, in the next step, nodes A, B and C make R as their parent and replicate this advertisement, however, by respectively changing ETX values to 1, 1.5, and 3. In the second step, node D receives advertisements from both A and C and computes its path ETX as follows:

$$\text{my ETX to neighbour } X + \text{ETX from } X \text{ to } R \tag{1}$$

Suppose both links $\overrightarrow{DA}$ and $\overrightarrow{DC}$ are of the same quality, D selects A as its immediate parent as the path over this node is clearly optimal. However, in this step C also receives the advertisement of A and realises that using a single hop to reach R is more costly in terms of ETX than using A as a relay node. Therefore, it selects node A as its new parent and uses the new ETX value for subsequent advertisements. This process continues with the hope that ETX of a link will not change dramatically and a stable tree will be established with all nodes in the network joining the tree by selecting their parents. Tree construction suffers from typical distance-vector routing pathologies such as count to infinity, loops, and stranded nodes (Tanenbaum, 2002). Routing protocols employ mechanisms to recover from such pathologies. For example, loops are detected if a packet exceeds the maximum allowed number of hops specified in the *time-to-live* field.

Figure 8 Tree construction example. The tree root R advertises itself with a distance of 0. Each node joins the tree by selecting a parent that minimises the remaining cost (such as ETX) to the tree root



3.1.2 Key properties

Now we define key properties to establish a base for a fair functional comparison of routing case studies. Most of these properties are similar to link estimation properties discussed in the previous section, however, their definitions are extended at the network level instead of just a node's one hop neighbourhood.

- *Stability*: Similar to link estimation, stability points to the steadiness of a routing topology and how gracefully a routing protocol recovers from node and link failures in the network.
- *Adaptability*: It determines how well a routing protocol adapts to the underlying link conditions, i.e., by responding to link estimator's suggestions, to enhance performance parameters such as throughput and number of transmissions.
- *Scalability*: It shows the maximum stretch of the routing topology in terms of how many nodes can be supported in the network without any communication breakdown. This is one of the most important properties for routing protocols in sensor networks because the envisioned scale of deployment surpasses thousands of cooperating networked objects (motes).
- *Reliability*: The delivery rate of a routing protocol. It is one of the most important measure of routing performance in multihop wireless networks.
- *Overhead*: Routing overhead is measured in terms of transmission, i.e., the frequency and size of routing update messages, and storage, i.e., the memory required for maintaining routing structures such as routing tables.

3.2 Case studies

From service point of view, we can divide routing protocols in wireless networks in two broad categories: *proactive* and *reactive*. As the name suggests, proactive routing protocols actively establish routing topology once a network is in place and a protocol is activated irrespective of if the applications really want to send data. Hence, such protocols maintain a connected network at any time. CTP (Gnawali et al., 2013), MintRoute (Hegazy et al., 2010), OLSR (Clausen and Jacquet, 2003) are among the examples of proactive routing protocols.

On the other hand, reactive routing protocols are demand based and only establish a route when two nodes intend to communicate with each other. Once the communication is over, the routes are typically disabled after a certain period of time. These protocols are specifically useful in challenging networking conditions and mobility scenarios when maintaining an active routing topology is costly. DSR (Johnson et al., 1996), AODV (Perkins et al., 2003) and DYMO (Billington and Yuan, 2009) are well known examples of reactive routing approaches.

We will now discuss three case studies:

- CTP, state-of-the-art proactive routing in sensor networks
- opportunistic routing (Biswas and Morris, 2005a, 2005b), a novel approach of exploiting link diversity in mesh networks
- AODV, a widely used reactive routing approach used both in sensor networks and MANETs.

3.2.1 Collection tree protocol

CTP (Gnawali et al., 2009) is one particular instance of collection tree protocol described in Rodrigo et al. (2006). It is state-of-the-art and one of the most widely used collection protocols shipped with TinyOS (Levis et al., 2004*a*; Levis and Gay, 2009), an OS platform for sensor networks implemented in nesC language (Gay et al., 2003a, 2003b). It uses 4BLE as its link estimator.

The basic operational principle of CTP is the same as distance vector based tree construction discussed earlier. However, it uses some novel mechanisms to address two common problems of distance vector based approaches;

- loops
- slow response to topological changes (Tanenbaum, 2002).

The former is clear, however, the later requires some explanation. In distance vector routing, any news (e.g., node addition or breakdown), spreads across a network very slowly, one hop per update interval. So a node cannot be fully incorporated or removed from routing decisions until the news has spread across the whole network. Decreasing the update interval is a straight forward solution to spread the news quickly, however, it generates unnecessary traffic which is prohibitive in energy constrained sensor networks. In order to address these problems, CTP introduces the following two mechanisms.[2]

Datapath validation: Typically, routing protocols use update messages to detect loops. However, CTP actively monitors data packets to solve any discrepancies along the data path. Each packet carries the transmitter's local ETX estimate to the destination, calculated using the mechanism discussed in Section 3.1.1. Logically, the ETX of the recipient node shall always be less than the ETX value in the received packet. This is because the transmitter will only send a packet to its parent that is closer to the destination than itself. A packet is considered to be in loop if it violates this rule, i.e., its ETX is less than or equal to the receiver's ETX. Consequently, the receiver node initiates data path validation instead of simply dropping the packet. The data path validation deals with updating the ETX estimates of an out-of-date node using *adaptive beaconing*.

Adaptive beaconing: As already mentioned, the sending frequency of routing updates (beacons) is a tradeoff between resource consumption and the recentness of the topology. CTP introduces an adaptive beaconing mechanism to strike an efficient tradeoff between the two. Using this mechanism, in emergency situations – such as addition/deletion of a node or loop detection – the network can respond within milliseconds by aggressive beaconing, while slowing it down significantly in normal conditions to save energy and bandwidth. The adaptive beaconing is a modification of Trickle (Levis et al., 2004b) algorithm used for disseminating code updates in the network. In Trickle, a node suppresses its update and doubles the update-interval if it overhears a similar update, or decreases the update-interval to the minimum when it receives a new code update. Similarly, adaptive beaconing mechanism expands or shrinks a node's beaconing interval based on stable or unstable topological conditions in a network, respectively.

Rating: Figure 9 evaluates CTP on functionality accounts. It is a very stable collection protocol based on long-term link estimation and efficiently repairs discrepancies in its routing topology. Therefore, we assign it four points for stability. The adaptivity of CTP stems from 4BLE: It changes a degrading link after just five failed transmissions. However, it is unable to use valuable opportunities on links that are black-listed by the link estimator due to their dynamic and bursty nature. Nonetheless, quick recovery of the topology using adaptive beaconing earns it two points.

Figure 9 The performance rating and use case for CTP

Stability	Adaptability	Scalability	Reliability	Overhead	Note
●●●●○	●●○○○	●●●●○	●●●●○	●●●○○	Proactive collection

CTP only maintains a constant number of neighbours, all one hop neighbours at maximum, in the routing table irrespective of the network size and node density. Therefore, it achieves high scalability (four points) as demonstrated by empirical evaluations in Gnawali et al. (2009). The reliability of CTP is well proven as it has been thoroughly tested on twelve testbeds using six different link layer protocols (Gnawali et al., 2009).

It delivered more than 90% of the packets on all testbeds with different physical topologies and varying link conditions, and hence, we assign it four points for its reliability. Finally, the overhead of CTP accounts for equation (1) the routing beacons exchanged among the neighbouring nodes using the adaptive beaconing mechanism discussed earlier,

$$\text{Tx. Overhead} = \text{BeaconSize} \times \text{BeaconFrequecy} \qquad (2)$$

and equation (2) the routing table, which maintains the state of a subset of one hop neighbours of a node. The default table size in CTP's implementation is restricted to 10 entries.

3.2.2 Opportunistic routing

Opportunistic routing (or ExOR) (Biswas and Morris, 2005a, 2005b), and its derivative ORPL (Duquennoy et al., 2013) for sensor networks, tries to exploit long range intermediate links for routing purposes. It tries hard to forward packets over intermediate links that offer better routing progress and are closer to the destination. However, after delivering 90% of the packets in a *batch*, it uses the reliable delivery mechanism of an underlying routing protocol, such as OLSR, for delivering the remaining packets over the traditional path.

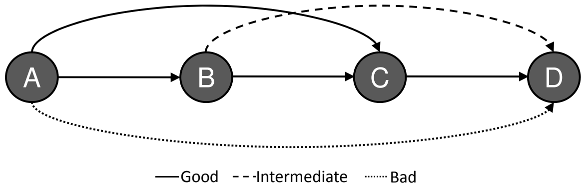
ExOR is based on the idea of cooperative diversity (van der Meulen, 1977) that uses broadcast transmissions to forward information through multiple relays. The destination can then use the best received signal or even combine information, i.e., to reconstruct the signal, received via multiple relays. ExOR utilises two unique opportunities of link diversity in multihop wireless networks. First, using broadcast packet transmissions, it utilises intermediate nodes along the traditional routing path to forward packets if the transmission falls short of the intended recipient. This way, the progress already made by a packet is utilised since an intermediate node, instead of the sender, forwards the packet further. Second, the packet may travel farther (e.g., 2 hop distance) than the intended recipient. ExOR makes use of this luck by providing mechanisms to allow farthermost recipient of the packet to become the next forwarder instead of the intended recipient.

Figure 10 explains the basic idea behind ExOR: Lets assume node A wants to send a packet to node D. In traditional routing, such as CTP, it forwards the packet to node C, the next hop in the routing table for node D. Suppose node C fails to receive this transmission but node B does. ExOR utilises this opportunity by allowing node B to deliver this packet either directly to node D or via its next hop. Similarly, in the second case, the transmission from node A might occasionally be received by node D directly. ExOR also allows the routing protocols to take advantage of this good fortune.

In the following, we discuss the three main operational ingredients of the opportunistic routing protocol.

Determining the forwarder set: ExOR determines a prioritised subset of nodes that shall be responsible for receiving and forwarding the packet. To compute the forwarder set, ExOR requires knowledge about the loss rate of each link in the network. In the first step, a sender node calculates the shortest path to the destination. The first node in this path gets the higher priority to forward packets. The same procedure is repeated to complete the forwarder set by deleting the previously selected candidates from calculations and assigning lesser priority to the node that is selected in the later step. This forwarder set is then cached until the next link-state update. Each packet contains its forwarder list in the header.

Figure 10 A simple example explaining the cooperative diversity utilised in opportunistic routing. Packets from node A to node C might occasionally be received by destination D directly or by node B. ExOR exploits such opportunities by avoiding retransmissions from node A

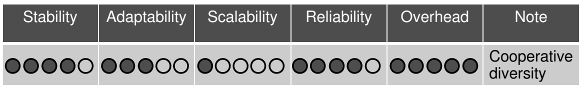


Agreement protocol: The nodes in the forwarder set then use an agreement protocol to forward the packet. ExOR operates on batches of packets to minimise the overhead of the agreement protocol. The main purpose of the agreement protocol is to schedule the time when a node should transmit its fragment of the batch. Higher priority nodes, as indicated by the forwarder set, are allowed to transmit first. A node maintains a *forwarder timer* that is scheduled far ahead to allow higher priority nodes to transmit first. This timer is readjusted when the node overhears other node's transmission. Each node also maintains a *batch map* that determines, for each packet, the highest priority node known to have received that packet. The agreement protocol heavily relies on packet overhearing to update batch maps.

Reliable delivery: ExOR does not offer reliable delivery. Therefore, it uses the traditional routing as a backup mechanism, which employs hop-by-hop acknowledgements, for delivering the lost packets requested by the destination.

Rating: The rating for ExOR is given in Figure 11. ExOR uses ETX based routing topology maintained by an underlying routing protocol. Therefore, we assign it four points for stability (as we did in the case of ETX based CTP's topology). Rating ExOR's adaptability is not straight forward: Although its performance is heavily dependent upon the underlying link condition, it does not pay any specific attention to varying link conditions at the routing layer. Nonetheless, it employs a highly efficient algorithm for packet forwarding that prioritises the next hop selection; with the node closest to the destination always being assigned the highest priority. In short, ExOR's algorithm ensures that every progress made by a packet during a single transmission is utilised without taking link dynamics directly into consideration (Adaptability = three points).

Figure 11 The performance rating and use case for ExOR



Stability	Adaptability	Scalability	Reliability	Overhead	Note
●●●●○	●●●○○	●○○○○	●●●●○	●●●●●	Cooperative diversity

ExOR doest not scale well because it needs link-state information of the whole network. Therefore, we assign it only one point for its limited scalability. ExOR itself does not guarantee reliable delivery. However, the use of traditional routing as a backup ensures that it is at least as reliable as the traditional routing itself. Hence, it is assigned four points for reliability. The biggest limitation of ExOR is the overhead associated with its agreement protocol that includes

- forwarder lists and batch maps appended with each transmitted packet
- packet overhearing to update node state
- the computation complexity of the protocol itself. Therefore,we assigned 5 points for its high overhead.

3.2.3 AODV

In Section 3.2.1, we discussed CTP that has been specifically designed to meet the stringent resource constraints of sensor networks. Similarly, in the previous section, we discussed ExOR that exploits wireless link diversity. AODV, on the other hand, is a more general reactive routing approach: It was originally designed for MANETs and later adapted to sensor networks. Reactive routing approaches are useful in challenging network conditions where maintaining a consistent routing topology is expensive. For example, in a network with mobile nodes (e.g., MANETs) or limited connectivity between nodes due to harsh environmental conditions (e.g., sensor networks). There are three steps in AODV's reactive routing approach,

- route request
- route reply
- route maintenance.

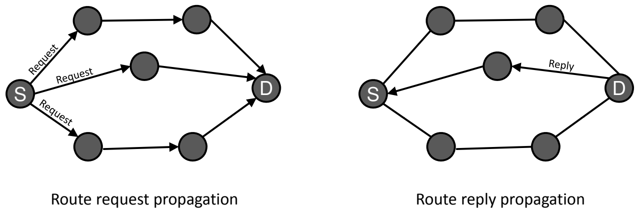
Route request: In AODV, each node maintains a small table containing information, such as a set of neighbours to forward packet to, for a particular destination. Links with neighbours are generally considered available or unavailable. Hence, it does not perform any active link estimation.

The route discovery in AODV proceeds as follows: When a source S wants to send a packet to destination D, it looks up its table to see if there is a neighbour entry for node D. Route discovery is only initiated if it does not find an entry in the table for node D. The same procedure is repeated at every intermediate node. When an entry for node D is not found, node S broadcasts a $ROUTE\_REQUEST$ packet relayed by all its neighbours until it reaches the destination or an intermediate node that already has an entry for node D in its table. Figure 12 explains this process by showing the paths taken by the $ROUTE\_REQUEST$ packet to reach node D.

Route reply: When a route aware intermediate node or destination D receives the request it replies with a $ROUTE\_REPLY$ packet. However, this $ROUTE\_REPLY$ packet is now unicasted along the same path over which it was received – the smallest path is chosen if multiple requests are received – in the opposite direction (cf. Figure 12). As this packet traverses through the network, each intermediate node records an entry for node D for to establish a distance vector. Once this packet is received by the source of $ROUTE\_REQUEST$ packet, it initiates its data communication with node D.

Route maintenance: Node mobility can cause sudden changes in the network topology. Therefore, a node has to keep track of which routes in the table are valid from time to time. In this regard, a node regularly exchange $HELLO$ messages, to which, each of its neighbour is suppose to respond. If a response message is not received, all the associate entries for the non responsive neighbours are deleted from the table.

Figure 12 Route request and reply propagation through the network in AODV

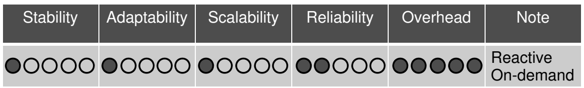


Route request propagation

Route reply propagation

Rating: AODV is a customised routing protocol only feasible in specific scenarios, i.e., networks with high node mobility. It does not match the stability of proactive routing approaches as routes are established based on simple route request and reply primitives. There is no link estimation performed and hence low quality links can dominate AODV's route selection without any particular consideration given to their feasibility for data transmission. Hence, it only receives one point for its stability as depicted in Figure 13. AODV treats links as available or unavailable, giving no consideration, whatsoever, to the underlying changes in link quality. It is assigned one point for its adaptability because of its route request and reply mechanism that allows for path reconstruction.

Figure 13 The performance rating and use case for AODV



Stability	Adaptability	Scalability	Reliability	Overhead	Note
●○○○○	●○○○○	●○○○○	●●○○○	●●●●●	Reactive On-demand

Longer tables with multiple forwarding candidates for each destination and an expensive route discovery mechanism strongly limit the scalability (1 point) of AODV. A number of studies (Gomez et al., 2006; Yang et al., 2008; Klein, 2008; Deshmukh and Ambhaikar, 2010; Che-Aron et al., 2010) have been performed to implement and evaluate AODV in different types of wireless networks under varying networking conditions. The reported reliability results of AODV differ significantly (e.g., from <50% (Klein, 2008) to >90% (Perkins et al., 1999) due to difference in evaluation environments. Similarly, the lack of link estimation makes it more susceptible to long range links of bad quality. The inclusion of such links results in frequent route discovery across the network due to frequent transmission failures on unreliable paths. Therefore, we assign it only two points for reliability.

Finally, it has a very high overhead (five points) both in terms of memory footprint and bandwidth consumption due to frequent exchange of HELLO messages and route requests. However, it is clear that AODV targets specific ad hoc communication scenarios with high node mobility.

4 Addressing

Point-to-point communications in multihop wireless networks require an addressing scheme to locate nodes in the network. Many addressing schemes have been proposed both for sensor networks and mesh networks such as geographical (Imieliński and Navas, 1999; Bose et al., 1999; Karp and Kung, 2000; Kuhn et al., 2003), hierarchical (Tsuchiya, 1988a, 1988b; Eriksson et al., 2007) and virtual coordinate addressing (Cao and Abdelzaher, 2006; Fonseca et al., 2005; Moscibroda et al., 2004; James and Song, 2003; Rao et al., 2003). However, our focus lies on self-configurable and decentralised addressing schemes which are equally relevant in multiple types of wireless networks. For example, it is not always feasible to deploy additional hardware in sensor networks to locate a node for routing purposes. Therefore, in this section, we concentrate on addressing schemes that derive virtual node locations based on the underlying connectivity in a network.

4.1 Introduction

There are two main ingredients of point-to-point communication in multihop wireless networks, addressing and routing. Addressing deals with assigning locations to nodes in the network topology. A far located sender node uses this address for routing purposes. Routing on the other hand deals with actual decision making at each node to select the best next hop for the packet to reach its destination. In general, routing is performed greedily to allow for a scalable communication infrastructure that only requires a node to know its one hop neighbourhood.

4.1.1 Challenges

Assigning locations to nodes in a multihop wireless network is a complicated task. As opposed to wired networks, there is no permanent network infrastructure that can be manually configured beforehand. Many factors contribute to rapidly changing topologies in a network such as node breakdown due to battery depletion in sensor networks and a large number of participants moving, leaving, or joining the network in MANETs.

Traditional addressing schemes, such as IP, greatly suffer if applied to multihop wireless networks. For example, IP based hierarchical addressing is not feasible because it requires a very careful manual configuration of the entire network assuming a static topology. Unlike wired network, an IP address of a node in a wireless network is merely used to identify a node in the network for Internet communications but it does not reveal the routable location of the node. Another solution is geographical addressing that also requires either manual configuration or GPS support. However, in wireless networks, the connectivity graph is dynamic and strongly depends on the physical environment. Therefore, a geographical path leading towards a node might not be the optimal path based on the connectivity graph. Similarly, geographic routing suffers heavily from holes (or dead ends (You et al., 2009; Leong et al., 2006) in the network.

In recent years, virtual coordinates based addressing schemes have received much attraction in the research community for two main reasons: First, they are completely decentralised and self configuring. It means that nodes determine their addresses themselves after joining the network without any central coordination or manual configuration. Second, these schemes are based or the underlying connectivity graph, and hence, a node's address guides the packets to follow the best path leading towards the node. These benefits of virtual coordinate-based addressing mechanisms make them suitable for both sensor networks and mesh networks. Before presenting a few case studies on virtual coordinate-based addressing schemes, we identify the key properties of an addressing scheme in a wireless network.

4.1.2 Key properties

Following are the key properties of an addressing scheme that we use to compare state-of-the-art case studies.

- *Address Stability*: This property states the number of times a node changes its address. Address changes may occur due to
    - variations in the underlying link conditions
    - frequent node failures.

  It is an important property because a node's location is typically stored in a distributed global database in the network and every change in the address requires an update in that database. Hence, address update is an expensive operation.
- *Address monotony:* Once an address change occurs, this property determines the magnitude of difference (e.g., in hop counts) between a node's previous and new location. A smaller change in address (i.e., high address monotony) could result in higher routing success even if the packets are routed towards the destination using its outdated addresses. This is because the packets may still reach the vicinity of the destination whose new location is very close to the old one.
- *Resilience*: It shows how well an addressing scheme recovers from frequent node additions and departures from the network. In such dynamic scenarios, a resilient addressing scheme would require far less address updates in the network than a non-resilient one.
- *Scalability*: This is similar to routing scalability in the previous section. It shows the ability of the addressing scheme to enlarge itself to accommodate the growing number of nodes in the network.
- *Overhead*: It is measured in terms of storage requirements and control packets, such as beacons or address updates in the global database, exchanged to maintain stable addressing in the network.

4.2 Case studies

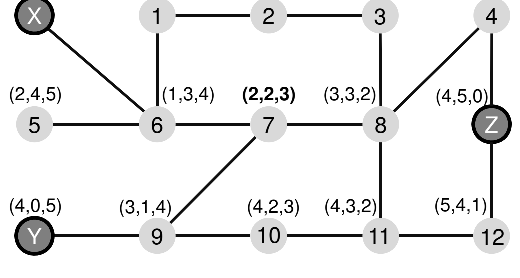
In this section, we present three well known case studies of point-to-point routing, namely BVR (Fonseca et al., 2005), S4 (Mao et al., 2010) and PAD (Alizai et al., 2011; Alizai and Wehrle, 2014). BVR is one instance of virtual coordinates based addressing specifically implemented for sensor networks. S4 is a cluster based extension of BVR that achieves significantly smaller routing stretch than BVR. PAD assigns probabilistic addresses to nodes instead of sharp virtual coordinates to improve the stability of virtual coordinate based addressing mechanisms.

4.2.1 Beacon vector routing (BVR)

BVR is also based on tree construction primitive. However, it needs multiple trees each rooted at *landmark*. A landmark is a designated node in the network used as a reference point by all other nodes. Every node in the network identifies its position in each landmark tree. The location of a node is defined in terms of a vector of hop distance from each landmark, commonly referred to as *virtual coordinates*. Routing is performed greedily over the virtual coordinates. There are two operational ingredients of BVR, virtual coordinate based addressing and routing.

Virtual coordinates: Figure 14 shows an example of BVR's virtual coordinate based address establishment in a network with three tree roots (landmarks). Landmarks advertise themselves by repeatedly sending beacons. Based on these beacons, each node S (recursively) determines the number of hops $h(S, L_i)$ to each landmark L_i. The result can be viewed as a set of routing trees with the landmarks as their roots and with, for example, the hop count as a routing metric. A node S's coordinates $\vec{c}(S)$ in the virtual coordinate system are the λ-dimensional vector $< h(S, L_1), \ldots, h(S, L_\lambda) >$ with λ as the total number of landmarks. In our example in Figure 14, node 7 has a three-dimensional address vector $< 2, 2, 3 >$ where each vector component represents the node's hop distance to the landmarks X, Y, and Z, respectively.

Figure 14 Virtual coordinates based addressing in BVR. Each node determines the hop distances from landmarks in the network. A vector of these hop distances, i.e., virtual coordinates, is used as a node's routable address



Routing: Routing is performed greedily over these addresses. The idea is to let a node S choose a next hop T that minimises the remaining distance $d(T, D)$ to the destination D (e.g., select a neighbour as a next hop whose coordinates are most similar to those of the destination node). BVR uses absolute component-wise difference as a routing metric:

$$d(T, D) = \sum_{i=1}^{\lambda} |T_i - D_i| \qquad (3)$$

However, real-world deployments are confronted with lossy links that may falsely influence the hop distance from landmarks. It means that traversing one hop can require more than one transmission. Therefore, the 'best' next hop is the one that results in the least number of transmissions necessary to reach the destination. BVR employs a link estimator to identify neighbours with stable links that minimises ETX for a successful delivery. Thus, only a selected subset of neighbours – offering an ETX below a certain threshold – are used in calculating the hop distance from the landmarks. Nonetheless, a node's address vector still represents the hop distance over the path with minimum ETX.

Rating: Figure 15 rates the performance of BVR. The address stability of BVR strongly depends upon the underlying network conditions. Tree construction offers a simple and attractive addressing solution, however, it is increasingly difficult to maintain stable trees in challenging network conditions. Changes in a particular node's coordinates propagate throughout the network and trigger further changes down the tree. For example, if a node close to a landmark changes its coordinate component for that landmark, all the descendant nodes will have to change their coordinates as well. Therefore, we only assign two points to BVR with regard to address stability.

Figure 15 The performance rating and use case for beacon vector routing

Stability	Monotony	Resilience	Scalability	Overhead	Note
●○○○○	●●○○○	●○○○○	●●●○○	●●●●○	Virtual coordinates

The magnitude of change in node's coordinates (address monotony) is calculated by summing the absolute component-wise difference of each coordinate component. The idea is to see if changes in a node's coordinates are sudden or gradual. As BVR's tree construction process is based on long term link estimation, it strongly limits the number of options for reaching landmarks and this usually results in a higher magnitude of change in addresses. For example, a node may change its hop distance from two to four (magnitude of change = two points) for a certain landmark because this is the best option available among the set of limited neighbours with high quality links. Hence, BVR is only assigned two points for its address monotony.

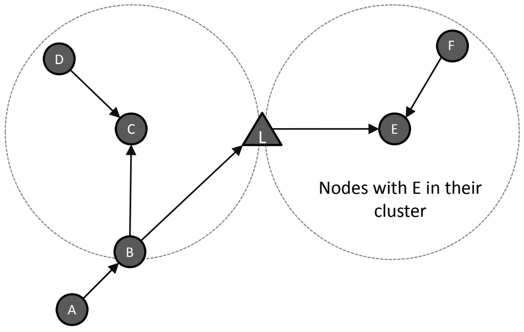
BVR is not particularly resilient to address changes because it is unable to locally recover from node additions or failures (Fonseca et al., 2005). Thus, node dynamics lead to significant changes in the topology throughout the addressing tree (resilience = one point). The scalability of BVR is comparable to any other tree construction based routing approach, such as CTP. However, the state maintained per node is not constant and depends upon the number of landmarks in a network (scalability = three points). Besides state maintenance at each node, BVR relies on expensive packet overhearing based link estimator that appends link estimation information with each outgoing packet (Overhead = four points).

4.2.2 Small state and small routing (S4)

S4 is a cluster based extension of BVR that significantly reduces routing stretch – the ratio of the hop count of selected path to that of the optimal path (Mao et al., 2010). S4 argues that the attempts to maintain small state per node to achieve higher scalability can result in undesirable routing performance in terms of routing stretch. It minimises both the state and routing stretch by combining the distance-vector based global network state and scoped distance-vector based local cluster state. S4 achieves an average routing stretch of 1 and is the state-of-the-art point-to-point routing protocol in sensor networks.

Algorithm: Apart from maintaining global virtual coordinates in the network, as in BVR, each node in S4 maintains a routing table for all the nodes in its local cluster. A node S's local cluster $C_k(S)$ contains all the nodes whose distance to D are within k times their distances to their closest landmarks. The idea behind maintaining a local cluster is that a node S can intercept packets addressed to node D and deliver them directly (cf. Figure 16). This significantly reduces the routing stretch because the packet does not have to reach the closest landmark of the destination, and then from there to the destination itself.

Figure 16 S4's routing scenarios. (1) A → C: B intercepts packets from A to deliver them directly to C instead of traversing through landmark L. (2) A → E: No shortcut is found and the packet is delivered via landmark (as in BVR). (3) D → C or F → E: Shortest path routing is used as the destinations are within the local cluster of sender nodes



Source: Mao et al. (2010)

One of the key advantages of S4 is its small addresses. As opposed to BVR, the routing approach of S4 does not require the whole coordinate vector to be included as destination's address in the packet header. For packet forwarding, a node's address is nothing but the ID of its closest landmark. If a node wants to send a packet to another node within its cluster, it directly forwards the packet to the destination over the shortest path. However, if a sender node from another cluster sends a packet towards the destination's closest landmark, the packet is either intercepted by an intermediate node with destination in its cluster or it finally reaches the landmark and then delivered to the destination. Figure 16 depicts multiple routing situations and shows how S4 reacts in each of these situations.

Optimisations: Maintaining two level topological structure requires robust mechanism both for maintaining both inter-cluster and intra-cluster topology. S4 introduces relevant mechanisms to ensure a stable topology at both levels. For inter-cluster routing each node is supposed to know its (shortest-path) distance to all the landmarks in the network. Therefore, a reliable delivery of beacon packets (advertisements) initiated by landmarks is necessary to maintain a stable topology. This is because sudden packet losses can sometimes result in miscalculation of the distance while other times may require substantial changes in the topology, thereby degrading the performance of S4. To address these challenges, S4 requires every node S in the network to rebroadcast beacons until n neighbours have received it or the maximum retransmission count t_{max} has reached. The choice of $t_{\max}$ and n is a tradeoff between the overhead and reliability.

Similarly, for intra-cluster routing, a node S has to retransmit a packet until an acknowledgment is received or the maximum retransmission count has been reached. In the later case, S initiates a local failure recovery request. After receiving this request, S's neighbours try to recover the packet locally. The idea is to select a node that is closest to the destination as the next hop for S. To avoid an explosion of local failure responses, in case a large number of S's neighbours maintain a distance vector for the destination in their tables, a prioritised response mechanism based on their distance vector is used. This way, each neighbour knows how long it has to wait before sending a failure recovery response.

Rating: The main difference between BVR and S4 is that the latter achieves smaller routing and transmission stretch. However, with regard to the properties defined for our functional comparison, S4 gathers a similar rating as BVR for scalability, monotony and resilience, as shown in Figure 17. Although S4 needs to maintain a global and local state per node, it can be as scalable as BVR by carefully selecting the state bounds. Similarly, address monotony remains the same because global virtual coordinates of S4 are based on BVR's distance vector approach. Recovering from node failures is at least as troublesome as in BVR. S4 achieves slightly higher stability than BVR because of its beacon rebroadcasting mechanism. Finally, the overhead of S4 is similar to BVR (Mao et al., 2010).

Figure 17 The performance rating and use case for S4 protocol

Stability	Monotony	Resilience	Scalability	Overhead	Note
●●○○○	●●○○○	●○○○○	●●●○○	●●●●○	Routing stretch

4.2.3 Probabilistic addressing

Virtual coordinates, such as in BVR, offer an attractive addressing mechanism for multihop wireless networks whose deployments are often unplanned and lack any permanent network infrastructure. However, their direct adoption of tree construction primitive is not as efficient as in address-free collection protocols (cf. Section 3.2.1). This is because in virtual coordinate addressing both addressing and routing are strongly coupled with each other: A change in a node's immediate parent does not only impact the routing path towards a tree root but also the routable location of that node and all its descendants. Hence, link quality changes along a tree branch (i.e., routing path) force virtual coordinate based addressing mechanisms to recompute addresses of all the nodes connected to the tree via that branch. This limitation strongly impedes the routing performance despite high overhead for regular address updates in challenging network conditions.

For example, in Figure 14, node 7's virtual location with respect to landmark Y will heavily rely on the path $7 \rightarrow 9 \rightarrow Y$. Each time a node changes its hop distance from a landmark, all child nodes have to modify their hop distances to that landmark as well. As a result, any node failure or changes in the quality of the links (due to data loss) on this path will not only trigger a change in the routing topology but also in the virtual coordinates (location in the network) of node 7.

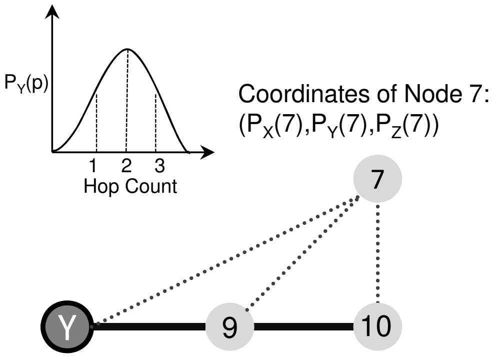
Probabilistic Addressing (PAD) (Alizai et al., 2011; Alizai and Wehrle, 2014) addresses these limitations of BVR and S4 by expressing a node's virtual coordinates in the form of probability distributions. For example, in Figure 18, node 7 can also reach landmark Y over the unreliable paths $7 \rightarrow Y$ and $7 \rightarrow 10 \rightarrow 9 \rightarrow Y$. PAD defines a node's location on the basis of all possible paths that can be used to reach the landmarks regardless of the estimated quality of these paths (cf. Figure 18), just like an orbital function describes possible quantum states of an electron around an atom (Daintith, 2008). Thus, it introduces a degree of fuzziness in a node's address that acclimatises short-term changes in the underlying link conditions. In PAD, the address of node S is a vector of frequency distributions of all paths qualities leading towards a landmark:

$$\vec{c}(S) = < F(S, L_1), \ldots, F(S, L_\lambda) > \tag{4}$$

where, $F(S, L_1)$ is the frequency distribution of path qualities of all the paths that can be used by node S to reach Landmark L_1. After calculating its new address $\vec{c}(S)$, S compares it to its previous address $\vec{c}'(S)$ by calculating a difference value d (e.g., via Pearson's χ^2-test). S only updates its address if d exceeds a certain threshold.

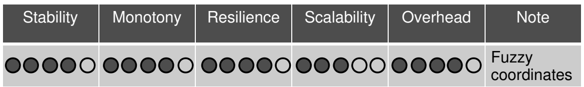
Hence, PAD doest not maintain any explicit trees in the network and automatically supports the inclusion of long-range intermediate links into the routing process by embedding information regarding multiple paths leading towards a node in its address distribution. Thereby, a node's topological location is no longer dependent on a particular path but on a subset of such paths. As a result, link quality changes along a single path does not necessarily change the location of the nodes along that path since these nodes are reachable over multiple paths. Routing on PAD uses the same distance function as in BVR (cf. Section 4.2.1). However, the distance function is calculated using a node's $mean$ coordinates (derived from PAD addresses) to select the best neighbour locally.

Figure 18 Probabilistic coordinates in PAD: Each node determines probability of hop distances from landmarks in the network



Rating: Figure 19 provides a comparative rating for PAD. One of the key advantages of PAD is its stability (four points). Because it assigns fuzzy address to nodes instead of sharp coordinates, PAD concedes a degree of *error* in its addresses achieving an order of magnitude higher address stability than BVR and S4.

Figure 19 The performance rating and use case for PAD



Stability	Monotony	Resilience	Scalability	Overhead	Note
●●●●○	●●●●○	●●●●○	●●●○○	●●●●○	Fuzzy coordinates

PAD achieves a very high address monotony (four points) because it is neither dependent on stringent tree like topology nor on expensive link estimation. Hence, unlike BVR, which depends upon robust parent selection predominantly influenced by routing cost metric and link estimation, PAD lessens the need for this stringent parent-child relationship in a network. A node in PAD does not have a static position but a region where it can reside just like an electron resides in its region around the nucleus of an atom. As long as a node is within its assigned region, it can be reached without needing to change its coordinates. The same reasons apply for address resilience: Since a node is no longer dependent on a single parent, a sudden departure of a node does not necessarily impact the location of its descendant. However, in BVR, the importance of a node grows with regard to address resilience depending upon

- how close it is to a landmark
- how many descendants it holds in the tree.

For example, the departure of a node closer to a landmark can break the whole routing tree associated with that landmark and inflict address changes throughout the network.

Despite its long addresses, PAD achieves similar scalability as BVR and S4. First, because it only maintains a subset of (useful) routes leading towards landmark in its address distribution regardless of the node density (Alizai and Wehrle, 2014). Second, because it offers a number of design choices with regard to address establishment, aggregation and dissemination in the network. For example, one such scalable design choice would be to aggregate PAD addresses in the form of *mean* or *weighted average* and use them for routing purposes. The main overhead of PAD is its long addresses. However, when compared with BVR and S4, PAD neither employs packet overhearing nor link estimation for establishing addresses in the network. Therefore, we assign it a similar rating (Overhead = four points).

5 Summary and conclusions

In this paper, we discussed link estimation, routing, and addressing concepts in multihop wireless networks. We also presented state-of-the-art case studies from each of these three areas and compared them based on a desired set of properties. Figure 20 summarises our comparison.

Figure 20 Summary of the comparative rating assigned to case studies in the area of link estimation, routing, and addressing

Link Estimation	Stability	Adaptability	Current link state	Reception correlation	Overhead
4BLE	●●●●●	●●○○○	●○○○○	●○○○○	●●●○○
SOFA	●○○○○	●●○○○	●●●●●	●●○○○	●●●●○
4C	●●●●●	●●●●○	●●●●●	●●●●○	●●●●○
Routing	Stability	Adaptability	Scalability	Reliability	Overhead
CTP	●●●●○	●●○○○	●●●●○	●●●●○	●●●○○
ExOR	●●●●○	●●●○○	●○○○○	●●●●○	●●●●●
AODV	●○○○○	●○○○○	●○○○○	●●○○○	●●●●●
Addressing	Stability	Monotony	Resilience	Scalability	Overhead
BVR	●○○○○	●●○○○	●○○○○	●●●○○	●●●●○
S4	●●○○○	●●○○○	●○○○○	●●●○○	●●●●○
PAD	●●●●○	●●●●○	●●●●○	●●●○○	●●●●○

Long term link estimation is the preferred mechanism employed by today's link estimators. Its primary goal is to establish a stable routing topology. However, in achieving this goal, it mainly disregards packet reception correlation and the current state of a link at the time of packet transmission. 4BLE adds a degree of adaptiveness to this estimation technique by demoting a link immediately after five consecutive packet failures. This helps in improving routing reliability but contributes little towards the goal of utilising long range intermediate links in the network. Contrarily, short term link estimation primarily focuses on the current link state but achieves limited stability – a primary routing design requirement. 4C tries to combine the advantages of both these approaches. It does not underestimate the need of stable routing topology, while at the same, provides relevant mechanisms to estimate long range intermediate links and utilise them for packet forwarding.

Routing protocols typically utilise high quality links for packet forwarding. The idea is to convert the network graph into a simplistic tree like structure and only use the links that form the branches of that tree. In doing so, they limit packet forwarding to a very limited set of links. ExOR provides an elegant solution to efficiently utilise link diversity and the broadcast nature of wireless medium. However, its computational requirements and reliance on link-state information for each directed link in the network limits its usage to resource rich platforms such as in mesh networks.

Virtual coordinate based point-to-point routing approaches are also unable to exploit link diversity to ensure long term stable addressing in the network. Despite a tremendous emphasis on address stability, these approaches suffer from frequent address updates in dynamic network conditions. In this regard, PAD provides a sophisticated solution to address both these problems. A PAD address is composed of multiple paths leading towards a node and also exposes the quality of these paths in the form of a probability distribution. Moreover, it assign fuzzy locations to nodes to account for sudden changes in link conditions and thus maintain stable addressing across the network even under challenging network conditions.

References

Aguayo, D., Bicket, J., Biswas, S., Judd, G. and Morris, R. (2004) 'Link-level measurements from an 802.11b mesh network', *SIGCOMM Comput. Commun. Rev.*, Vol. 34, pp.121–132.

Alizai, M.H. and Wehrle, K. (2014) 'Probabilistic location-free addressing in wireless networks', *Journal of Network and Computer Applications*, Vol. 41, No. 0, pp.167–181.

Alizai, M.H., Landsiedel, O. and Wehrle, K. (2012) 'Exploiting the burstiness of intermediate quality wireless links', *International Journal of Distributed Sensor Networks (IJDSN)*, Vol. 2012, Article ID 826702, 17 p.

Alizai, M.H., Landsiedel, O., Link, J. A.B., Götz, S. and Wehrle, K. (2009) 'Bursty traffic over bursty links', *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems', SenSys '09*, ACM, New York, NY, USA, pp.71–84.

Alizai, M.H., Vaegs, T., Landsiedel, O., Goetz, S., Link, J.A.B. and Wehrle, K. (2011) 'Probabilistic addressing: Stable addresses in unstable wireless networks', *IPSN 2011: Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*, Chicago, IL, USA, IEEE, pp.61–72.

Alizai, M.H., Wirtz, H., Kirchen, B. and Wehrle, K. (2013) 'Portable wireless-networking protocol evaluation', *Journal of Network and Computer Applications*, Vol. 36, No. 4, pp.1230–1242.

Becher, A., Landsiedel, O., Kunz, G. and Wehrle, K. (2008) 'Towards short-term link quality estimation', *Proceedings of the International Workshop on Hot Topics in Embedded Networked Sensing Systems, HotEmnets '08*, Charlottesville, Virginia, USA, ACM, pp.1–5.

Billington, J. and Yuan, C. (2009) 'On modelling and analysing the dynamic manet on-demand (dymo) routing protocol', *T. Petri Nets and Other Models of Concurrency*, Vol. 3, pp.98–126.

Biswas, S. and Morris, R. (2005a) 'Exor: opportunistic multi-hop routing for wireless networks', *SIGCOMM Comput. Commun. Rev.*, Vol. 35, pp.133–144.

Biswas, S. and Morris, R. (2005b) 'Exor: opportunistic multi-hop routing for wireless networks', *Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM '05*, ACM, New York, NY, USA, pp.133–144.

Bose, P., Morin, P., Stojmenović, I. and Urrutia, J. (1999) 'Routing with guaranteed delivery in ad hoc wireless networks', *Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, DIALM '99*, ACM, New York, NY, USA, pp.48–55.

Cao, Q. and Abdelzaher, T. (2006) 'Scalable logical coordinates framework for routing in wireless sensor networks', *ACM Trans. Sen. Netw.*, Vol. 2, pp.557–593.

Cerpa, A., Wong, J.L., Kuang, L., Potkonjak, M. and Estrin, D. (2005a) 'Statistical model of lossy links in wireless sensor networks', *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks, IPSN '05*, IEEE Press, Piscataway, NJ, USA.

Cerpa, A., Wong, J.L., Potkonjak, M. and Estrin, D. (2005b) 'Temporal properties of low power wireless links: Modeling and implications on multi-hop routing', *Proc. of the 6th ACM Interational Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Urbana-Champaign, IL, USA, ACM, pp.414–425.

Che-Aron, Z., Al-Khateeb, W. F.M. and Anwar, F. (2010) 'The enhanced fault-tolerant aodv routing protocol for wireless sensor network', *Proceedings of the 2010 Second International Conference on Computer Research and Development, ICCRD '10*, IEEE Computer Society, Washington DC, USA, pp.105–109.

Clausen, T. and Jacquet, P. (2003) *Optimized Link State Routing Protocol (OLSR)*, RFC 3626, IETF.

Daintith, J. (2008) *Oxford Dictionary of Chemistry*, Oxford University Press, 6th ed., April, Oxford, UK.

De Couto, D.S.J., Aguayo, D., Bicket, J. and Morris, R. (2005) 'A high-throughput path metric for multi-hop wireless routing', *Wirel. Netw.*, Vol. 11, pp.419–434.

Deshmukh, R. and Ambhaikar, A. (2010) 'Performance evaluation of aodv and dsr with reference to network size', *International Journal of Computer Applications*, Vol. 11, No. 8, pp.27–32, Published By Foundation of Computer Science.

Duquennoy, S., Landsiedel, O. and Voigt, T. (2013) 'Let the tree bloom: scalable opportunistic routing with ORPL', *SenSys'13: Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, Rome, Italy, ACM, pp.1–14.

Eriksson, J., Faloutsos, M. and Krishnamurthy, S.V. (2007) 'Dart: dynamic address routing for scalable ad hoc and mesh networks', *IEEE/ACM Trans. Netw.*, Vol. 15, pp.119–132.

Fonseca, R., Gnawali, O., Jamieson, K. and Levis, P. (2007) 'Four-bit wireless link estimation', *Sixth Workshop on Hot Topics in Networks (HotNets)*, Atlanta, Georgia, USA, ACM, pp.1–7.

Fonseca, R., Ratnasamy, S., Zhao, J., Ee, C.T., Culler, D., Shenker, S. and Stoica, I. (2005) 'Beacon vector routing: scalable point-to-point routing in wireless sensornets', *Proceedings of the 2nd conference on Symposium on Networked Systems Design and Implementation - Volume 2, NSDI'05*, USENIX Association, Berkeley, CA, USA, pp.329–342.

Gay, D., Levis, P., von Behren, R., Welsh, M., Brewer, E. and Culler, D. (2003a) 'The nesc language: a holistic approach to networked embedded systems', *Proceedings of the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation, PLDI '03*, ACM, New York, NY, USA, pp.1–11.

Gay, D., Levis, P., von Behren, R., Welsh, M., Brewer, E. and Culler, D. (2003b) 'The nesc language: a holistic approach to networked embedded systems', *SIGPLAN Not.*, Vol. 38, pp.1–11.

Gnawali, O., Fonseca, R., Jamieson, K., Kazandjieva, M., Moss, D. and Levis, P. (2013) 'Ctp: an efficient, robust, and reliable collection tree protocol for wireless sensor networks', *ACM Trans. Sen. Netw.*, Vol. 10, No. 1, pp.16:1–16:49.

Gnawali, O., Fonseca, R., Jamieson, K., Moss, D. and Levis, P. (2009) 'Collection tree protocol', *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, SenSys '09*, ACM, New York, NY, USA, pp.1–14.

Gomez, C., Salvatella, P., Alonso, O. and Paradells, J. (2006) 'Adapting aodv for ieee 802.15.4 mesh sensor networks: Theoretical discussion and performance evaluation in a real environment', *Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks, WOWMOM '06*, IEEE Computer Society, Washington DC, USA, pp.159–170.

Hedrick, C.L. (1988) *Routing Information Protocol*, 1988 RFC Editor, USA.

Hegazy, I., Safavi-Naini, R. and Williamson, C. (2010) 'Towards securing mintroute in wireless sensor networks', *Proceedings of the 2010 IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), WOWMOM '10*, IEEE Computer Society, Washington DC, USA, pp.1–6.

Huang, M.L. and Park, S-C. (2009) 'A wlan and zigbee coexistence mechanism for wearable health monitoring system', *Proceedings of the 9th International Conference on Communications and Information Technologies, ISCIT'09*, IEEE Press, Piscataway, NJ, USA, pp.555–559.

Imieliński, T. and Navas, J.C. (1999) 'Gps-based geographic addressing, routing, and resource discovery', *Commun. ACM*, Vol. 42, pp.86–92.

James, N. and Song, D. (2003) 'Gem: Graph embedding for routing and data-centric storage in sensor networks without geographic information', *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, SenSys '03*, ACM, New York, NY, USA, pp.76–88.

Johnson, D.B., Maltz, D.A. and Broch, J. (2001) 'DSR: the dynamic source routing protocol for multihop wireless ad hoc networks', *Ad Hoc Networking*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, pp.139–172.

Karp, B. and Kung, H.T. (2000) 'Gpsr: greedy perimeter stateless routing for wireless networks', *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, MobiCom '00*, ACM, New York, NY, USA, pp.243–254.

Kim, T.H., Ha, J.Y. and Choi, S. (2009) 'Improving spectral and temporal efficiency of collocated ieee 802.15.4 lr-wpans', *IEEE Transactions on Mobile Computing*, Vol. 8, pp.1596–1609.

Klein, A. (2008) 'Performance comparison and evaluation of AODV, OLSR and SBR in mobile ad-hoc networks', *International Symposium on Wireless Pervasive Computing (ISWPC)*, Santorini, Greece, p.5.

Kuhn, F., Wattenhofer, R., Zhang, Y. and Zollinger, A. (2003) Geometric ad-hoc routing: of theory and practice', *Proceedings of the Twenty-Second Annual Symposium on Principles of Distributed Computing, PODC '03*, ACM, New York, NY, USA, pp.63–72.

Landsiedel, O., Ghadimi, E., Duquennoy, S. and Johansson, M. (2012) 'Low power, low delay: Opportunistic routing meets duty cycling', *IPSN'12: Proceedings of the 11th ACM/IEEE International Conference on Information Processing in Sensor Networks*, Beijing, China, ACM, pp.185–196.

Lee, S-B., Kwak, K.J. and Campbell, A.T. (2006) 'Solicitation-based forwarding for sensor networks', *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on*, Vol.1, pp.90–99.

Leong, B., Liskov, B. and Morris, R. (2006) 'Geographic routing without planarization', *Proceedings of the 3rd conference on Networked Systems Design and Implementation – Volume 3, NSDI'06*, USENIX Association, Berkeley, CA, USA, pp.25–39.

Levis, P. and Gay, D. (2009) *TinyOS Programming*, 1st ed., Cambridge University Press, New York, NY, USA.

Levis, P., Madden, S., Gay, D., Polastre, J., Szewczyk, R., Woo, A., Brewer, E. and Culler, D. (2004a) 'The emergence of networking abstractions and techniques in tinyos', *NSDI'04: Proceedings of the 1st Conference on Symposium on Networked Systems Design and Implementation*, USENIX Association, Berkeley, CA, USA, pp.1–14.

Levis, P., Patel, N., Culler, D. and Shenker, S. (2004b) 'Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks', *Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation – Volume 1*, USENIX Association, Berkeley, CA, USA, pp.2–15.

Liu, T. and Cerpa, A.E. (2011) 'Foresee (4c): wireless link prediction using link features', *Proceedings of the 10th International Conference on Information Processing in Sensor Networks, IPSN '11*, ACM, New York, NY, USA.

Mao, Y., Wang, F., Qiu, L., Lam, S. and Smith, J. (2010) 'S4: small state and small stretch compact routing protocol for large static wireless networks', *IEEE/ACM Trans. Netw.*, Vol. 18, pp.761–774.

Moscibroda, T., O'Dell, R., Wattenhofer, M. and Wattenhofer, R. (2004) 'Virtual coordinates for ad hoc and sensor networks', *Proceedings of the 2004 joint workshop on Foundations of mobile computing, DIALM-POMC '04*, ACM, New York, NY, USA, pp.8–16.

Niculescu, D. (2007) 'Interference map for 802.11 networks', *Proceedings of the 7th ACM SIGCOMM conference on Internet Measurement, IMC '07*, ACM, New York, NY, USA, pp.339–350.

Perkins, C., Belding-Royer, E. and Das, S. (2003) *Ad Hoc On-Demand Distance Vector (aodv) Routing*, 2003 RFC Editor, USA.

Perkins, C.E., Royer, E.M. and Das, S.R. (1999) 'Ad hoc on-demand distance vector (aodv) routing', *2nd IEEE Workshop on Mobile Computing Systems and Applications*, February, New Orleans, LA, pp.90–100.

Puccinelli, D. and Haenggi, M. (2006) 'Multipath fading in wireless sensor networks: measurements and interpretation', *Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing, IWCMC '06*, ACM, New York, NY, USA, pp.1039–1044.

Puccinelli, D. and Haenggi, M. (2008) 'Duchy: double cost field hybrid link estimation for low-power wireless sensor networks', *Proceedings of Fifth Workshop on Embedded Networked Sensors (Hot EmNets'08)*, Charlottesville, Virginia, USA, ACM, pp.1–6

Qiu, L., Zhang, Y., Wang, F., Han, M.K. and Mahajan, R. (2007) 'A general model of wireless interference', *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking, MobiCom '07*, ACM, New York, NY, USA, pp.171–182.

Rao, A., Ratnasamy, S., Papadimitriou, C., Shenker, S. and Stoica, I. (2003) 'Geographic routing without location information', *Proceedings of the 9th annual International Conference on Mobile Computing and Networking, MobiCom '03*, ACM, New York, NY, USA, pp.96–108.

Rappaport, T. (2001) *Wireless Communications: Principles and Practice*, 2nd ed., Prentice-Hall PTR, Upper Saddle River, NJ, USA.

Rodrigo, F., Gnawali, O., Jamieson, K., Kim, S., Levis, P. and Wo, A. (2006) *The Collection Tree Protocol*, TinyOS Enhancement Proposal, TEP 123.

Srinivasan, K., Jain, M., Choi, J.I., Azim, T., Kim, E.S., Levis, P. and Krishnamachari, B. (2010) 'The κ-factor: Inferring protocol performance using inter-link reception correlation', *Proceedings of the 16th Annual International Conference on Mobile Computing and Networking (Mobicom 2010)*, Chicago, Illinois, USA, ACM, pp.317–328.

Srinivasan, K., Kazandjieva, M.A., Agarwal, S. and Levis, P. (2008) 'The β-factor: measuring wireless link burstiness', *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems, SenSys '08*, ACM, New York, NY, USA, pp.29–42.

Tanenbaum, A. (2002) *Computer Networks*, 4th ed., Prentice-Hall Professional Technical Reference, Prentice-Hall Inc., Upper Saddle River, New Jersey, USA.

Tsuchiya, P.F. (1988a) 'The landmark hierarchy: a new hierarchy for routing in very large networks', *SIGCOMM Comput. Commun. Rev.*, Vol. 18, pp.35–42.

Tsuchiya, P.F. (1988b) 'The landmark hierarchy: a new hierarchy for routing in very large networks', *Symposium Proceedings on Communications Architectures and Protocols, SIGCOMM '88*, ACM, New York, NY, USA, pp.35–42.

van der Meulen, E. (1977) 'A survey of multi-way channels in information theory: 1961-1976', *Information Theory, IEEE Transactions on*, Vol. 23, No. 1, pp.1–37.

Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, J. and Alexander, R. (2012) *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*, RFC 6550 (Proposed Standard).

Woo, A., Tong, T. and Culler, D. (2003) 'Taming the underlying challenges of reliable multihop routing in sensor networks', *SenSys '03: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, ACM, pp.14–27.

Yang, T., Ikeda, M., Marco, G.D., Barolli, L., Durresi, A. and Xhafa, F. (2008) 'Routing efficiency of aodv and dsr protocols in ad-hoc sensor networks', *Proceedings of the 2008 The 28th International Conference on Distributed Computing Systems Workshops*, IEEE Computer Society, Washington, DC, USA, pp.66–71.

You, J., Lieckfeldt, D., Reichenbach, F. and Timmermann, D. (2009) 'Context-aware geographic routing for sensor networks with routing holes', *Proceedings of the 2009 IEEE Conference on Wireless Communications and Networking Conference, WCNC'09*, IEEE Press, Piscataway, NJ, USA, pp.2589–2594.

Zhang, H., Arora, A. and Sinha, P. (2009) 'Link estimation and routing in sensor network backbones: Beacon-based or data-driven?', *IEEE Transactions on Mobile Computing*, Vol. 8, pp.653–667.

Zhao, J. and Govindan, R. (2003) 'Understanding packet delivery performance in dense wireless sensor networks', *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, SenSys '03*, ACM, New York, NY, USA, pp.1–13.

Zhu, T., Zhong, Z., He, T. and Zhang, Z-L. (2010) 'Exploring link correlation for efficient flooding in wireless sensor networks', *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation, NSDI'10*, USENIX Association, Berkeley, CA, USA, pp.4–17.

Notes

[1] All the nodes determine their routing addresses and link estimates cooperatively (in-network) without any manual configuration such as in IP and Geographic routing.

[2] These mechanisms of CTP have been incorporated in the standard IPv6 routing protocol, i.e., RPL (Winter *et al.*, 2012), for low power and lossy networks.