修士論文
Master's Thesis

DTN 型静的センサーネットワークにおける
遅延特性改善手法

Improving Data Delivery Latency in
DTN-based Static Sensor Network

平成 27 年 2 月提出

指導教員 江崎 浩 教授

東京大学院
情報理工学系研究科 電子情報学専攻

48-136456
Manussanun Buranachokphaisan

Abstract

Wireless sensor network (WSN) will be able to deliver several interesting or exciting applications e.g. environmental monitoring, smart home, smart city, military surveillance and so on. However, it is widely recogized that the performance of WSN is often degraded by unreliable and unpredictable quality of their wireless links. Some applications, for instance monitoring and surveillance applications, generally require high reliability on data delivery among sensors in any scale of the networks. Since many applications are delay tolerant but are error sensitive, one of the feasible methods to provide acceptable communication reliability in WSN is Delay/Disruption Tolerant Network (DTN). We call this network as *DTN-based sensor network*.

Unlike traditional WSN routing schemes, DTN uses hop-by-hop reliable data transmission with store-and-forward mechanism to assure high reliability over intermittently-connected network. In the previous work, it was demonstrated that Potential-based Entropy Adaptive Routing (PEAR) protocol, which is one of DTN-based routing protocol, can achieve high reliable communication quality and good scalability, over 50 node-scale testbed. However, large latency for data delivery in this system may not be preferable quality in many monitoring applications.

This thesis contributes the following three points for PEAR system. Firstly, we examined the behavior of DTN-based routing protocol and we showed that the results from the previous experiment with PEAR have large latency on data delivery for many monitoring applications. Our study reveals that the dynamic nature of wireless links is a major factor to increase the latency of data delivery. Wireless links sometimes become unidirectional and low quality leading to the increase of message losses. Although hop-by-hop reliable transfer is able to achieve reliable data delivery, unawareness of those links causes large delivery latency.

Secondly, we propose Farther-Aim-Shorter-Try (FAST) forwarding scheme to improve delivery latency of DTN-based sensor network. FAST is the integration of reliable DTN approach and traditional WSN routing schemes.

FAST modifies PEAR's current next-hop selection scheme to avoid the selection of unidirectional links and combines hop-by-hop store-and-forward mechanism with the traditional routing schemes, i.e. aware of link quality and introduction of data retransmission against the data loss.

Finally, the proposed method was implemented and evaluated on WiFi-based UTMesh testbed with various network sizes and conditions. FAST was evaluated on 16 node-floor scenario and 33 node-multistory scenario and compared its performance with PEAR. The experimental results show that FAST decreased median of data delivery latency by 64% in floor scenario and 85% in multistory scenario. FAST also achieved better performance than conventional retransmission schemes that retransmit messages to the same next-hop, in various node densities and message sizes.

Acknowledgement

I would like to express my sincere gratitude to Professor Hiroshi Esaki for giving me a great opportunity to be a part of his laboratory and a chance to study and conduct the interesting research. His priceless advices and guidance are greatly appreciated. He also allowed me to participate in many reserach activities where I can enhance useful knowledge for my research.

I am extremely grateful to Assistant Professor Hideya Ochiai for all of the valuable advices and kind supports during my master course. He always gives kind consideration to every student. He also gives me motivations and inspirations for progressing research. Thank you for giving me a chance to grow up more scientifically.

I really appreciate Assistant Professor Hirochika Asai and Assistant Professor Manabu Tsukada for their precious suggestions on my research and this thesis.

I would like to thank all members in Esaki Laboratory that always give kind supports and advices. I have had a great research life in Japan because I joined this laboratory.

Finally, I would like to express all my sincere gratitude to my family for all supports and encouragements. Without them, none of this would have been possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Wireless sensor network (WSN) is an emerging technology widely used in many types of applications such as environmental monitoring, healthcare, smart home, smart city, industrial automation, military surveillance and so on. WSN consists of distributed, wirelessly enabled embedded devices capable of employing a variety of sensors called nodes. Each node is equipped with one or more sensors, microcontroller, wireless transceiver, and energy source. Communication between source node and sink or destination (We use sink and destination alternately) is not always direct. It can proceed in a multi-hop fashion, so every node is also a relay node.

WSN have many advantages over traditional sensing technology. The first and notable feature is its low cost. WSN negates the need for costly wiring between nodes. In addition, relatively inexpensive microcontrollers and transceivers also cut the cost of implementation. Further, autonomously self-organized manner makes WSN easy to deploy and flexible. Nodes automatically initialize communication with other nodes in range and create ad-hoc mesh network for relaying data. This allows nodes to be deployed in any location with little configuration. However, the notorious shortcoming of WSN is time-varying, unreliable, unstable and unpredictable quality of wireless links that leads to message loss during the delivery. Message loss becomes worse when a message is delivered through more hops resulting in unscalability. In some applications, message loss is not a critical issue, but in monitoring and surveillance applications, the completeness of data is necessary for further analysis and prediction. For example, the sequence of events is crucial to decide the next proper operation in industrial automation [1] and the reliability should be more than 99% in smart grid applications [2]. There are many methods to assure reliability in WSN. Since many applications are delay tolerant but are error sensitive, one of the feasible methods to provide acceptable communication reliability is Delay/Disruption Tolerant Network (DTN) [3]. This type of network is known as *DTN-based sensor network*.

DTN describes the network which the communication links are expected high end-to-end delay (or delivery latency), low data rate or frequently disrupted. The link may be disrupted occasionally due to radio frequency interference, or predictably disconnected owing to motion and low-duty-cycle operation. End-to-end connection does not always exist, thus these networks are known as *Intermittently Connected Networks (ICNs)*. Popular examples of DTN scenarios include interplanetary network, underwater acoustic communication, mobile ad-hoc network (MANET), vehicular ad-hoc network (VANET) and so on. To guarantee the reliability in such networks, DTN relies on hop-by-hop reliability with *store-and-forward* mechanism instead of the traditional ARQ (Automatic Repeat reQuest) with retransmission. An intermediate node stores receiving messages in its storage (or buffer) for long period of time and periodically exchanges information with its next-hop to confirm that the messages are delivered. Most DTN routing protocols assume mobile nodes in the network, so node mobility is exploited to find the proper next-hop which is commonly the node that is close to or moves toward the sink.

WSN for monitoring applications differs from DTN, thus some approaches showing good performance in DTN result bad performance in WSN. Seeing that the nodes in monitoring applications are mainly stationary, the communication network is static multi-hop WSN. The works in Ref. [4] presented that even in static multi-hop wireless network, the connectivity was intermittent. Thus, they tried to apply DTN-based routing protocol to this network. This work demonstrated that the Potential-based Entropy Adaptive Routing (PEAR) protocol, which is one of DTN-based routing protocol, achieved 100% delivery rate over 10 hops over 50 node-scale wireless mesh network. This results show that DTN routing protocol was able to provide high reliability and scalability over the intermittently-connectivity. However, the results exhibit bad performance on delivery latency. The average and 99^{th} percentile delivery latency was 238s and 700s respectively. Such large latency for data delivery may not be preferable quality in many monitroing applications.

1.2 Objective

The objective of this research is to minimize delivery latency of DTN-based sensor network while preserving high reliability and scalability as in the previous work. This work focuses on the monitoring scenario which is a static multi-hop wireless network. The traffic pattern is many-to-one communication where all nodes periodically send data to one destination.

Indeed, delivery latency constraint is difficult to determine because acceptable delivery latency depends on the applications. In this research, we indicate that the delivery latency should be less than the sensing interval. For example, if a sensor senses and sends a message every one minute, the message should be delivered to the destination within one minute as well.

1.3 Contributions

The contributions of this research are as follows:

1. Firstly, we analyze the problems of DTN-based routing protocol regarding the delivery latency in static multi-hop WSN. The analysis is proved by the results from the experiment on DTN-based sensor network in the previous work. We found that the dynamic nature of wireless links is the major cause of large delivery latency.

2. We propose the new forwarding scheme called *Farther-Aim-Shorter-Try (FAST)* to reduce delivery latency in DTN-based static sensor network. FAST is the integration of DTN and WSN routing approaches. FAST inherits hop-by-hop store-and-forward mechanism from DTN to assure the reliability, in addition to, introduces link quality metric and retransmission scheme, which are the traditional approaches in WSN, to forwarding scheme.

3. The proposed method was implemented and evaluated on UTMesh testbed [5] with different scales, deployments and conditions. The experimental results show that the proposed method significantly improved the delivery latency compared with PEAR. Moreover, it also outperformed traditional retransmission schemes.

1.4 Outline

This thesis consists of 6 chapters. This first chapter already introduced the motivation, objective and contributions of this research. Chapter 2 defines the problems of DTN-based static sensor network and describes related work. The proposed method is presented in Chapter 3 and the evaluation is shown in Chapter 4. Discussion is given in Chapter 5. Finally, Chapter 6 concludes this thesis.

Chapter 2

Problem Statement

This chapter defines the problems of DTN-based sensor network that lead to high delivery latency in monitoring applications. We first give background knowledge on wireless network and DTN routing approach. Then, we address design requirement for routing protocol in our interested network. After that, the problems are analyzed and verified by the results from the previous experiment on DTN-based sensor network. Finally, related work is presented.

2.1 Background

2.1.1 Intermittently-Connected Mesh Network (ICMeN)

Although WSN has several benefits above traditional networks, wireless communication is infamous for unreliability, instability and unpredictability. In an aspect of spatial characteristics, the transmission range can be seperated into three regions (Figure 2.1):

1. *Connected region*: Links are highly connected, good quality, stable and symmetric.

2. *Transitional region or gray area*: Links are frequently disrupted, intermediate quality, unstable, not correlated with distance and asymmetric.

3. *Disconnected region*: Links are low quality and more disconnected than connected.

Links in transitional region have special features which greatly affect on the routing performance. Most of links in the network falls into this

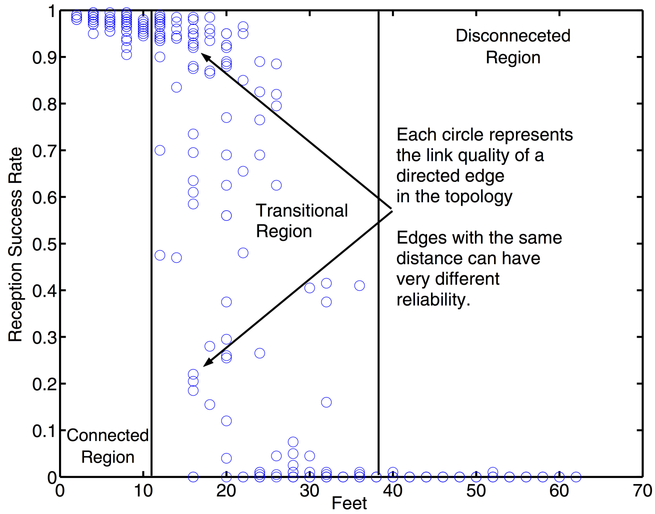


Figure 2.1: Three regions of wireless links. [6]

region. They are extremely variable and unpredictable. The average link quality in this region is intermediate, but the individual link exhibits high variation over time even nodes are immobile. Further, link quality is not correlated with the distance. Short links occasionally have poor connectivity, while distant links have excellent connectivity. Asymmetric links are also common in this region. Severe asymmetry leads to unidirectional links. Link asymmetry may or may not be persistent. It might be transient for unstable links, and ultimately depends on the environment, interference and radio hardware characteristic.

Due to wireless link characteristics in the transitional region, the connectivity of links in the network changes frequently even nodes are stationary. This kind of network is known as *Intermittently-connected Mesh Networks (ICMeN)* [4]. ICMeN is composed of stable wireless nodes, however, the links among them are disruptive and unreliable. They sometimes become connected, but also frequently become disconnected. Link availability is heterogeneous. Few links are tightly connected. The connected and disconnected duration of links cannot be predictable. The more hop count, the more packet loss. It is worth noting that ICMeN is different from ICN in terms of node mobility. Nodes in ICN could be stationary or mobile, while nodes in ICMeN are only stationary. As a result, our interested network falls into ICMeN.

2.1.2 Delay/Disruption Tolerant Network (DTN)

Delay/Disruption Tolerant Network (DTN) was originally introduced for the *challenged network* where the communication links are expected high latency, low data rate or frequently disrupted. End-to-end connection may not exist, therefore, DTN routing scheme must provide reliable transfer even in such networks.

Routing strategies

In general, a DTN routing protocol replicates the messages and delivers multiple copies of each message to the destination rather than selecting the best path and sending a single copy from node to node as a classical routing protocol does. Although this scheme notoriously wastes network resources because it creates multi-path from source to destination, it gives less delivery latency and greater delivery rate.

The other difference between DTN routing protocol and the classical one is that the former relies more on the mobility of mobile nodes to find time-varying topology and creates opportunity for routing under the assumption of intermittent connectivity. Most of existing protocols usually route the messages to the node that is close to or moves toward the destination.

Hop-by-hop reliable transfer

The Internet uses Transmission Control Protocol (TCP) which is a connection-oriented protocol to assure end-to-end reliability. The connection must be established before sending a message and maintained until the transmission finishes. Automatic Repeat reQuest (ARQ) is utilized to handle delayed, damaged or lost message. ARQ needs the response from the destination after sending each message. If the source node does not receive acknowledgement (ACK) from the destination, the message will be retransmitted from the source node. Seeing that TCP requires end-to-end continuous connectivity and instant end-to-end feedback, TCP is not preferable for DTN.

Instead of TCP, hop-by-hop transfer with *store-and-forward* mechanism was proposed to guarantee the delivery in DTN. An intermediate node has a storage (or a buffer) to store messages delivered through it. If there is no link or path to the destination, the intermediate node holds the messages until the link exists. ARQ is an optional in DTN since DTN expected large delay and link disruption. DTN node may provide *custody transfer* [7] which promises not to delete the messages in the buffer until they can be reliably delivered to another node providing custody transfer or they arrive at the destination. In custody transfer, ACK is sent back to the previous node to indicate that the messages are successful forwarded. The messages are retransmitted from the closest node rather from source in case of message lost. This approach provides reliable message transfer without complete contemporaneous end-to-end path to the destination.

A DTN node periodically (e.g. when two nodes come into the transmission range of one another) exchanges buffer information with another neighbour node as illustrated in Figure 2.2. Each node maintains list of messages in the buffer called *summary vector*. This vector is exchanged first, then only those messages absent in the other's summary vector are exchanged. With this protocol, each node knows whether the messages in its buffer are delivered to other nodes. This protocol is known as *epidemic exchange* since it was originated by Epidemic routing [8], or *investigation* due to its behaviour [9].

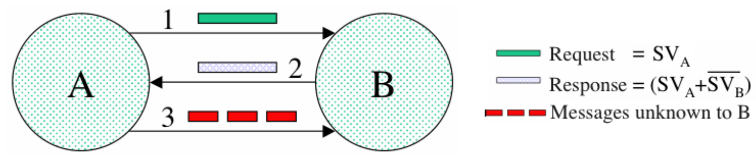


Figure 2.2: Epidemic exchange. [8]

2.2 Design Requirement

The design of routing protocol is influenced by many factors [10]. These factors are classified as network characteristic, application requirement and general requirement. Routing protocol is application-dependent i.e. different application has different requirement for routing protocol. Here, we explain some factors that we have considered in aspect of monitoring applications.

Network Characteristic

- *Traffic pattern*: Traffic pattern in WSN can be categorized as time-driven (continuous), event-driven and query-driven. Traffic in monitoring applications is time-driven delivery model which sensor nodes periodically sense and transmit the data of interest at constant periodic time intervals.

- *Network dynamics*: WSN may contain both stationary and mobile sensor nodes. Sensor nodes in monitoring applications are mainly stationary.

- *Node Deployment*: Node deployment defines the network topology and, hence, affects on the routing performance. The deployment can be deterministic or randomized. Most nodes in monitoring applications are deterministic since they can manually deployed at specific location. The decision of sensors' location should be based on transmission range, signal attenuation and link disruption which caused by interference.

- *Connectivity*: WSN could be dense or sparse depending on node deployment. High node density prevents the network from being isolated, on the contrary, low node density frequently leads to network partitioned. Nonetheless, the connectivity is intermittent even nodes are stationary and, consequently, the network topology is variable even in highly connected networks.

Application requirement

- *Scalability*: The number of sensors depend on the application. Home monitoring system may contain less than hundred of sensors. In contrast, large building monitoring system may consist of thousands of sensors. Thus, routing protocol should be able to scale under many sizes of networks and allow additional sensor nodes that might be added in the future.

- *Quality of Service*: Reliable communication network is very significant for monitoring applications. The sequence of events is essential in decision and prediction. Some data must be delivered in real-time or near real-time. Thus, WSN should deliver complete set of data to the destination within the reasonable delivery latency.

General requirement

- *Resource constraint*: A sensor node in WSN is resource constraint in terms of energy and memory. Some routing protocols require routing table leading to memory problem in large-scale network. Besides, a DTN node needs buffer space to store receiving messages. Multi-hop communication introduces high overhead (control message) for topology management and also consumes high energy. Consequently, routing protocol that minimize overhead and utilized memory is preferable.

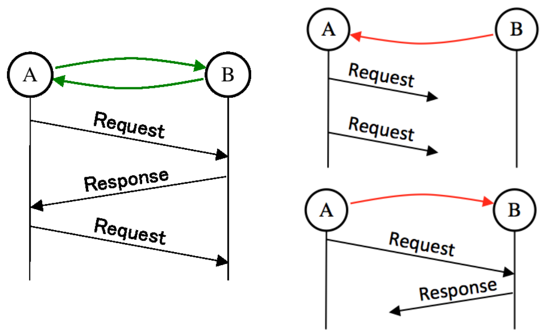
- *Fault Tolerance*: Some sensor nodes might fail or be blocked owing to physical damage or environmental interference. The failure may affect the overall task of the network. Routing protocol must provide formation of new routes to the destination.

2.3 Problem Analysis

In regard to network characteristic and DTN routing protocol, we analyze the problems leading to high delivery latency as follows,

1. Unidirectional links

Notice that the DTN node sends data messages only after it receives summary vector from the other node, a bidirectional link is necessary in order to succeed information exchange (Figure 2.3a). DTN assumes that links are always bidirectional. In reality, especially in static WSN, an unidirectional link possibly arises between two connected nodes when only one of two nodes can directly send the messages to the other. It may be caused by transmitter/receiver heterogeneity, power control of sensor node, interference or hidden terminal. If the unidirectional link turns up between a node and its next-hop, the request or response summary vector cannot be delivered (Figure 2.3b) and the messages, consequently, are not forwarded to the next-hop.



(a) Bidirectional link (b) Unidirectional link

Figure 2.3: DTN node's information exchange when the link between node A and node B is bidirectional and unidirectional.

2. Periodic retransmission

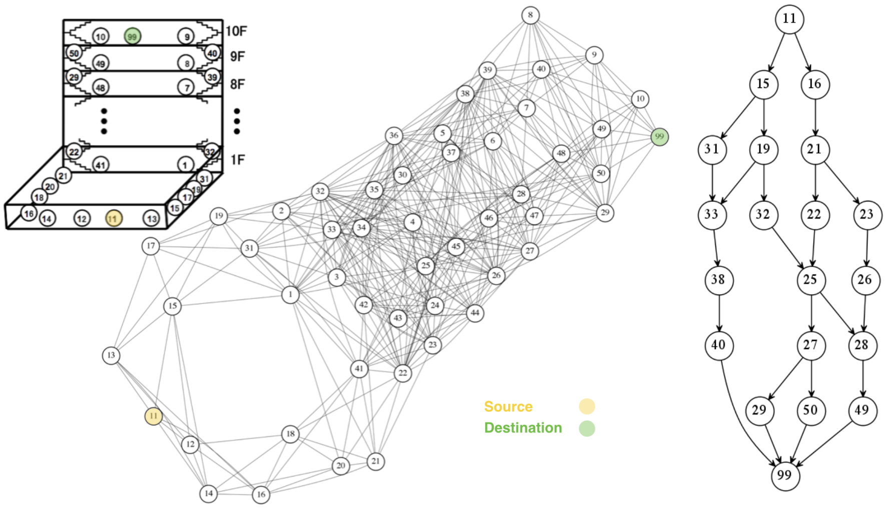
The traditional routing protocol uses Automatic Repeat reQuest (ARQ) with retransmission mechanism to ensure the reliability and reduce delivery latency. If the sender does not receive an acknowledgement (ACK) within a retransmission timeout, the sender retransmits the messages instantly until the sender receives ACK or reaches the maximum number of retransmission. In DTN, the retransmission is just an optional since DTN expects large latency. DTN relies on only the periodic transmission and acknowledgement. When the sender does not receive the response summary vector or fails to forward the data messages, the sender has to wait for next transmission time. This delays the message one transmission period.

3. Routing metric

Most DTN routing schemes are similar to hop metric in static multi-hop WSN where nodes select the least hop path in the delivery. Choosing the next-hop based on the number of hops gives low delivery latency in wired network or network containing mobile nodes, but this is not true for static WSN [11, 12]. Long links often have low quality and are easily disrupted due to signal attenuation and interference resulting in high probability of message drop. Both summary vector and data messages could be dropped during the delivery. Relating to the previous problem, nodes have to wait for next transmission time causing one period delivery latency.

2.4 Preliminary Study

We prove our assumptions by analyzing the results from experiment on DTN-based sensor network in Ref. [4]. Authors implemented and evaluated the performance of Potential-based Entropy Adaptive Routing (PEAR) protocol on wireless mesh network which is similar to our interested network. PEAR is a DTN-based routing protocol which inherits store-and-forward mechanism and hop-by-hop exchange to ensure message delivery. PEAR defines hop-by-hop exchange as *Investigation* process. A sender sends a request message to investigate a next-hop's buffer and forwards data messages only when it receives a response message. PEAR periodically investigates and transmits messages in the buffer to neighbour nodes. It updates network information and finds delivery path by periodically broadcasting advertisement message (ADV). An outstanding feature of PEAR is that PEAR does not target to specific mobility model, thus PEAR can be applied to both mobile and static network. However, PEAR has an aspect of distance-vector routing giving the minimum hop path in static network.



(a) Deployment & Topology (b) Delivery path

Figure 2.4: Deployment, topology and delivery path of the experimented network.

2.4.1 Experiment's details

Authors conducted the experiment with 51 nodes for 3.5 hours in Eng. Bldg. 2 in the University of Tokyo. The deployment configuration, topology and connectivity of the network is shown in Figure2.4a. All nodes were deployed on the floor statically (no mobile node). Node 11 on the 1^{th} floor is configured to send 100-byte messages to node 99 on the 10^{th} floor every 5s. Each node was working with Armadillo-220, a Linux embedded computer, with a WiFi module and operating at same transmitted/received power level. ADV broadcast interval, next-hop selection interval and retransmission interval were set to 10s. Therefore, nodes waited for next retransmission which caused 10s delay when they failed in forwarding the messages.

We first examined the delivery path from source to destination showing in Figure 2.4b. Although nodes chose only one next-hop at a time, nodes could have many next-hops during the experiment. Due to the intermittently-connected links, the next-hop was changed at each time slot. Thus, some nodes in the delivery path had more than one next-hops. Next, we observed investigation success rate of both nodes and links. The success of the investigation indicates if nodes forwarded the data messages to the next-hop. Finally, investigation success rate was compared to link characteristic to verify our assumptions.

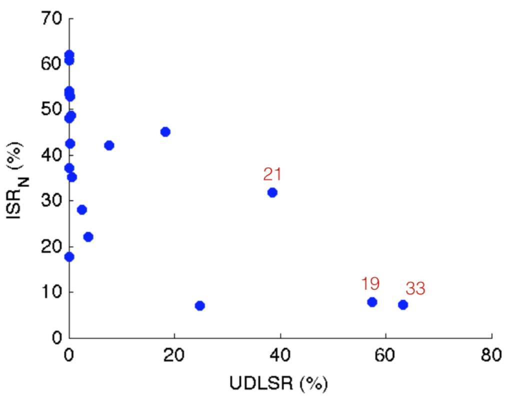
2.4.2 Effect of unidirectional links

The effect of unidirectional links on delivery latency was proved by comparing UniDirectional Link Selection Ratio ($UDLSR$) and Investigation Success Rate of each *node* (ISR_N). $UDLSR$ and ISR_N are defined in Equation (2.1) and Equation (2.2):

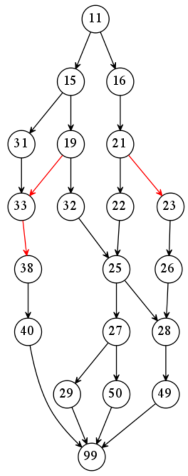
$$UDLSR = \frac{T_{udl}}{T_{exp}} \tag{2.1}$$

$$ISR_N = \frac{N_{res\_recv}}{N_{req\_send}} \tag{2.2}$$

where T_{udl} is the amount of time a node selects the next-hop with the unidirectional link, T_{exp} is the total experiment time, N_{res_recv} is the number of response messages a node receives from its next-hop and N_{req_send} is the number of request messages a node sends to its next-hop.



(a) The relationship between $UDLSR$ and ISR_N



(b) Unidirectional link in the delivery path.

Figure 2.5: The results showing the effect of unidirectional link.

Figure 2.5a shows the relationship between $UDLSR$ and ISR_N where each dot represents a node on the delivery path. We can see that nodes with high $UDLSR$ or nodes that regularly selected the next-hops with the unidirectional links were likely to fail in investigation. This result also corresponded to the link characteristic on the delivery path as illustrated in Figure 2.5b. Unidirectional links frequently arose between node 9, 21 and 33 and their next-hops and these nodes, hence, got higher $UDLSR$ than other nodes. Node 21 had intermediate ISR_N since it selected the next-hop with the unidirectional link only 40% of the experiment time. However, node 19 and node 33 suffered from unidirectional links approximately 60% of the experiment time. Therefore, the unidirectional link is one of the causes of high delivery latency.

However, some nodes in the left side of Figure 2.5a still got low ISR_N, even though they rarely selected the next-hops with unidirectional links. That means there were other factors causing low ISR_N.

2.4.3 Effect of low quality links

The effect of low quality links was observed in order to prove that hop metric always chooses low quality links in the delivery. In this research, link quality is simply estimated by calculating Advertise Reception Ratio (ARR) which defined in Equation (2.3). Since the investigation requires a bidirectional link, we multiply ARR on both directions to find bidirectional link quality denoted by $BiARR$ as in Equation (2.4).

$$ARR_{k \to n} = \frac{N_{adv\_recv}(n,k)}{N_{adv\_send}(k)} \tag{2.3}$$

$$BiARR(n,k) = BiARR(k,n) = ARR_{k \to n} \times ARR_{n \to k} \tag{2.4}$$

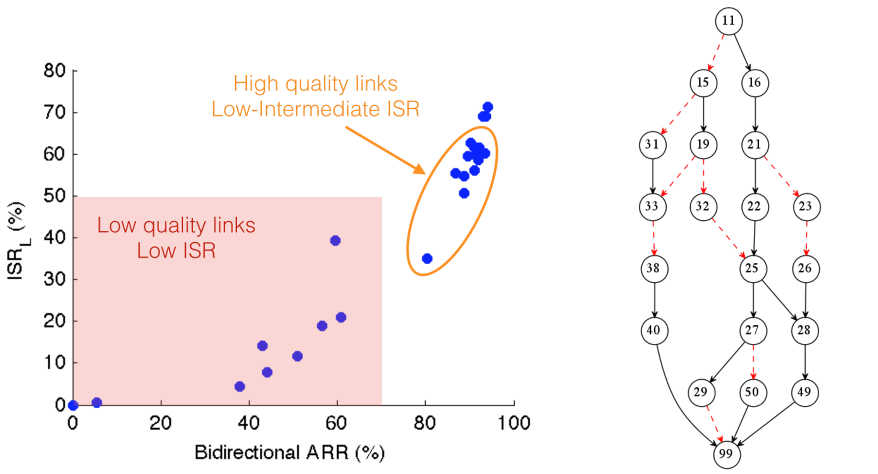
where $ARR_{k \to n}$ denotes ARR of the link from node k ($\in nbr(n)$) to node n, $N_{adv\_recv}(n,k)$ is the number of ADVs from node k received at node n, $N_{adv\_send}(k)$ is the number of ADVs sent by node k.

Then, $BiARR$ was compared to investigation success rate on each link (ISR_L) which defined in Equation (2.5).

$$ISR_L(n,k) = \frac{N_{res\_recv}(n,k)}{N_{req\_send}(n,k)} \tag{2.5}$$

where $ISR_L(n,k)$ is investigation success rate of the link that points out from node n to node $k \in nexthop(n)$, $N_{res\_recv}(n,k)$ is the number of response messages node n receives from node k and $N_{req\_send}(n,k)$ is the number of request messages node n sends to node k.

Figure 2.6a presents the relationship between $BiARR$ and ISR_L where each dot represents a link on the delivery path. The quality of links on delivery path was various. More than half of the links were high quality, however, there were still many low quality links (showing in the red area of Figure 2.6a). As we expected, ISR_L was proportional to $BiARR$. The low quality links resulted low ISR_L. Dash lines in Figure 2.6b illustrate low



(a) The relationship between $BiARR$ and ISR_L

(b) Low quality link in delivery path

Figure 2.6: The results showing the effect of low quality link.

quality links in the red area. The unidirectional link also includes in low quality link as one direction of the links had almost zero ARR. Considering nodes connecting with low quality links, some nodes chose another next-hop to forward the messages at times, but some nodes mainly chose neighbours with low quality links. Although node 19 had two main next-hops, it suffered from both unidirectional link and low quality link. Therefore, most of the messages delivered through these links were delayed. As a result, low quality link is another cause of high delivery latency.

2.4.4 Effect of periodic investigation

According to Figure 2.6a, some links still got low or intermediate ISR_L (40-60%) even those links had high $BiARR$ (as shown in the orange circle). Due to notoriously unreliable wireless links in the transitional region, the messages could be lost anytime though the quality of links were high. Since PEAR periodically retransmits the message in the buffer, any message lost causes one period delay. Messages were possibly lost continuously depending on the environment at the transmission time resulting in many periods delayed. Therefore, only periodic retransmission may not be able to give satisfied delivery latency.

2.4.5 Summary

The discovered problems were verified by the experimental results from the previous work. Even though investigation (hop-by-hop exchange) was able to guarantee high reliability over intermittently-connected link, unawareness of unidirectional and low quality links leaded to investigation failure and, consequently, high delivery latency. Further, periodic retransmission delayed message delivery in static multi-hop WSN when any message was lost. In this research, we aim to minimize delivery latency by improving the next-hop selection scheme according to these problems.

2.5 Related Work

In this section, related work on routing protocol in WSN and DTN is presented. Regarding discovered problems the last section, related work on unidirectional link and retransmission scheme is also described here.

2.5.1 Routing protocols in WSN

Many routing protocols have been proposed so far [10, 13]. Among these routing protocols, the most suitable one for monitoring applications is *Potential-based routing protocol* (PBR) [14] . This kind of protocol is also known as *Gradient-based routing protocol* [15] and *Utility-based routing protocol* [16].

Position awareness improves the efficiency and scalability of routing protocols as it helps reducing the number of messages used for route discovery. Position could be obtained by GPS, but it is not practical in WSN as it increases both cost and energy consumption. PBR was developed to overcome this problem by establishing virtual coordinate from scratch without external input. The purpose of such coordinate is not to mimic real geographic location, but to be used for feasible routing solutions. Each node in PBR has a scalar value called *potential* which represents the virtual distance from the sink. Nodes in the network create a *potential field* as the virtual coordinate by periodically broadcasting advertisement packet. Potential calculation relies on the requirement of applications e.g. hops, energy overhead, packet delivery rate and end-to-end delay. When receiving a message, nodes select the lowest-potential neighbour to forward the message. Figure 2.7 depicts the illustration of PBR when the potential field is established based on the number of hops from the destination. The number represents potential value of each circle.

The concept of PBR is particularly useful for monitoring applications which is a convergecast network. In the case that all messages are sent

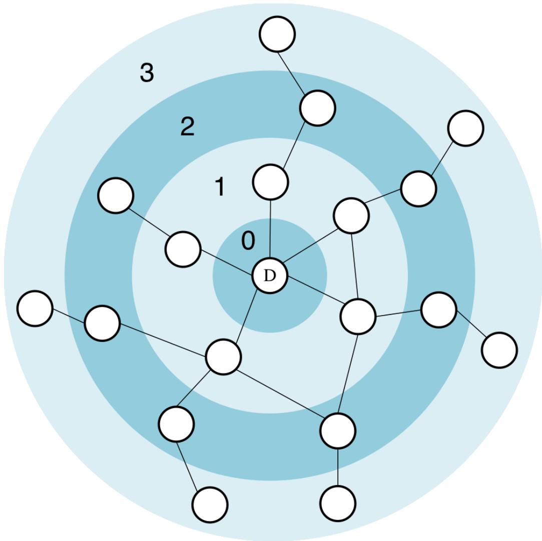


Figure 2.7: The illustration of PBR.

to a single sink, a single potential field that rooted at the sink is built and maintained in the network. PBR is energy efficiency, scalable, fault tolerance and adaptive to variable topology [17]. These advantages are correspond to monitoring application requirement, thus PBR is mostly used in monitoring applications.

Traditionally, WSN sends messages based on the shortest path or minimum-hop route as in wired network and DTN. Many studies found that hop metric tends to choose long and lossy links resulting in bad performance in static multi-hop WSN and recommended to be aware of path quality instead [11, 12].

A number of link quality metrics have been proposed in the past decades. There are metrics straightforwardly related to latency such as per-hop round trip time [18] and packet pair delay [11]. They use active unicast probes and ACK to measure latency and select path with minimum sum of latency in delivery. However, those metrics perform poorly due to self-interference [19]. Required number of packet transmissions-based (RNP) metric (e.g. ETX [20]) is the most widely used in WSN routing protocols. It approximates the total number of transmission packets required from source to destination. This metric also uses active probes and ACK in estimation. In the proposed method, we fix the number of retransmission to prevent congestion. Thus, this metric does not meet our requirement.

In our work, we try to enhance investigation success rate so that the sender has higher chance to transmit data messages. We seek the path with high delivery probability, so link quality is simply estimated using packet reception ratio as in [21]. To reduce overhead of control messages, our estimator exploits broadcast feature of PBR to approximate delivery probability by observing the number of receiving advertisement messages.

2.5.2 Routing protocols in DTN

Epidemic routing [8] is an early routing protocol proposed for DTN. It is the first protocol that defined hop-by-hop information exchange before forwarding the messages. Nodes spread the stored messages to every encountered node, thus this routing gives high delivery rate and low end-to-end delay due to the multi-path. However, it assumes that each node has infinite storage and bandwidth, so it is notoriously redundancy and high resource consumption. Later DTN protocols were developed based on epidemic routing, but they try to reduce redundancy by limiting number of replicated messages [22, 23].

Various routing metrics for minimizing end-to-end delay in DTN, such as Minimum Expected Delay (MED) and Earliest Delay (ED), have been presented by Jain et al. [24]. However, these metrics requires global knowledge and future connectivity of the network. It lacks of scalability and practicability due to the unpredictable connectivity of wireless links. Minimum Estimated Expected Delay (MEED) [25] is one of the popular routing protocols in DTN. MEED assumes that links are always bidirectional and have constant bandwidth and latency. MEED uses the observed connectivity history and assumes that the future connectivity will be similar to the previously observed connectivity. MEED's assumptions are not true for static WSN, in additional, the past connectivity sometimes cannot be used to predict future connectivity owing to the intermittent connectivity. There are other works [26, 27] proposed protocol for maximizing delivery rate and minimizing delivery latency, but they exploit the mobility characteristic in routing. Thus, they cannot apply to this work.

Potential-based Entropy Adaptive Routing (PEAR) protocol [9] is a DTN-based routing protocol which differs from the others in that it does not adhere to particular mobility pattern. Messages are replicated and delivered depending on node connectivity, so PEAR can be applied to static WSN as well. PEAR executes the investigation process before forwarding messages to the next-hop in order to confirm message delivery. PEAR modifies information in summary vector so that it does not guarantee only hop-by-hop delivery, but also assure end-to-end delivery implicitly. PEAR makes use of potential-based routing protocol to assign the position of nodes. PEAR outperformed other DTN routing protocols in several mobile environments and achieved high reliability and scalability in the static network.

PEAR gives many benefits to monitoring applications, except delivery latency. In DTN routing terms, node position is defined by the number of hops from destination. PEAR becomes distance-vector routing giving the minimum hop path in static multi-hop WSN. Our proposal takes advantage of hop-by-hop reliable transfer and modifies next-hop selection to avoid the problems mentioned in Section 2.3.

2.5.3 Unidirectional link

Real-world deployments of WSN revealed that link quality varies terribly over space and time [6, 29, 30], in addition to, indicated a significant presence and an effect of unidirectional links to the routing protocols [12, 31, 28]. Each existing research defined how to (1) detect and (2) handle unidirectional links.

Methods of detecting unidirectional links can be classified to estimation, common neighbour and acknowledgement techniques. Existing estimation techniques [32, 33] were developed based on Ad-hoc On Demand Distance Vector (AODV) routing [34] which is a reactive routing protocol using Route Request (RREQ) and Route Reply (RREP) to find path from source to destination on-demand. A node receiving RREQ estimates distance or expected signal strength of RREQ sender based on information included in RREQ such as transmit power, total noise, and minimum received power threshold. After that, it compares the estimated value with its capability and decides if the link is unidirectional. The evaluation in the existing research was done on the simulation only. Moreover, this technique has to estimate every time a node receives RREQ. It is not appropriate for proactive routing like PBR because it consumes a lot of resources for computation. The second technique discovers unidirectional links by using common neighbour [35]. Each node listens to all neighbours' beacons to detect their link quality. Nodes will volunteer to help neighbours relaying their discovery when they find asymmetric links. This technique works well only in a dense network and when a neighbour with good quality link exists. In additional, it consumes resources in computation because nodes calculate link quality of all links that they can hear beacons. The last method detects unidirectional links by acknowledgement (ACK). ACK is either explicit or implicit. AODV uses explicit ACK. RREQ sender knows that the link is unidirectional when it does not receive RREP. DEAL [36] detects asymmetric links with implicit ACK. Every node includes its neighbour table and neighbours' link quality in the beacon. When a node receives beacon, it knows that the link is unidirectional if it cannot not find itself in the neighbour table. Even though this method introduces more overhead than the others, we choose this method in our proposal since it is simple and any node can detect unidirectional links by itself without any condition.

When nodes discover unidirectional links, node choose to either avoid or exploit unidirectional links in routing. AODV and estimation techniques add neighbours with unidirectional links to a *blacklist set* for a period of time. When nodes receive RREQ from the sender in the blacklist set, nodes ignore RREQ and do not send RREP back to the sender. In contrast, the work in Ref. [10] and DEAL explain that the key factor of successful message delivery is forward link, so they exploit asymmetric link if the quality of forward link is high. Although their results prove that their works can improve the performance of typical WSN, this method cannot work with hop-by-hop reliable transfer. The bidirectional link is required in the investigation, so our proposal avoids the unidirectional links.

2.7.4 Retransmission

Retransmission with acknowledgement mechanism is a common approach for enhancing transmission reliability in both wired and wireless network. Retransmission scheme either performs on hop-by-hop or end-to-end basis. Analytical and simulation-based evaluation revealed that hop-by-hop retransmission performed better than end-to-end retransmission in terms of delivery latency and energy consumption [37].

The acknowledgement mechanisms are categorized into explicit acknowledgement (eACK), negative acknowledgement (NACK) and implicit acknowledgement (iACK). eACK and NACK send ACK packet to notify the sender whether the receiver receives the packet successfully or requests the missing packets, while iACK embeds ACK with other packets e.g. advertisement. iACK reaches a better performance than eACK and NACK, i.e. a greater reliability rate, with less delivery latency, overhead, energy and collisions [38, 39].

All retransmission schemes retransmit the messages to the same next-hop. Sometimes, the sender has to retransmit the messages many times before the messages reach the next-hop. Consequently, retransmission could aggravate the congestion in the network [40].

Our proposal derives the investigation from PEAR to promise 100% delivery rate. The investigation has already used iACK to confirm both hop-by-hop and end-to-end delivery. Our proposal makes use of retransmission with eACK during the investigation, but it limits the number of retransmission to alleviate congestion and provides alternative next-hop to avoid repeatedly message loss in retransmission.

Chapter 3

FAST Forwarding Scheme

We propose Farther-Aim-Shorter-Try (FAST) forwarding scheme to solve high delivery latency problem in DTN-based sensor network for monitoring applications. We first describe the overview of FAST. Each mechanism is explained in detail.

3.1 Overview and Design Principles

Farther-Aim-Shorter-Try (FAST) forwarding scheme integrates DTN approach and traditional WSN routing schemes to decrease delivery latency of DTN-based sensor network. The concept of FAST is to move the responsibility of forwarding messages to any neighbours closer to sink within one transmission period. FAST exploits the advantages of several mechanisms. It acquires PBR which is the most suitable protocol in monitoring applications. FAST inherits store-and-forward mechanism from DTN and conducts investigation before forwarding the messages to provide reliable communication on data delivery, but FAST applies retransmission to handle message loss during investigation. FAST also modifies PEAR's next-hop selection to avoid unidirectional links and prepares more reliable next-hop for retransmission.

Ideally, the route with lower number of hops gives lower delivery latency. Thus, FAST attempts to minimize the number of hops with hop metric first. A sender challenge sending the request message to a *primary next-hop* which is the farthest neighbour in the transmission range. When the sender fails in the investigation, i.e. does not receive a response message, the sender makes a reinvestigation once by retrasmitting the request message to an *alternative next-hop* chosen by link quality metric in order to prevent repeatedly failure. Thanks to high quality links, the sender has higher probability of successful investigation, as well as, forwarding the messages to another node closer to the sink. FAST gives only one chance for reinvestigation, thus the sender waits for next transmission period if it still fails in reinvestigation. The overview of FAST forwarding scheme is shown in Figure 3.1.

FAST makes use of PEAR's potential field construction to find a potential value of each node and uses the potential value to select the primary next-hop. Nodes periodically update their potentials with neighbours by broadcasting 1-hop advertisement. The node that is located farther from the sink has larger potential. FAST also manages replica and buffer as PEAR does. PEAR modified epidemic exchange to guarantee both hop-by-hop and end-to-end delivery. A node maintains the copy of each message until it receives certificate from the destination. The detail of potential field construction and replica & buffer management is explained in Appendix A.

FAST detects unidirectional links and estimates link quality by observing receiving ADVs. In this research, we call link and path quality metric as *Forward Predictability* and *Delivery Predictability* since they just predict whether a node has a chance to forward the data messages. Unidirectional link detecting technique and link quality estimation is discussed in Section Section 3.3 and 3.4 respectively.

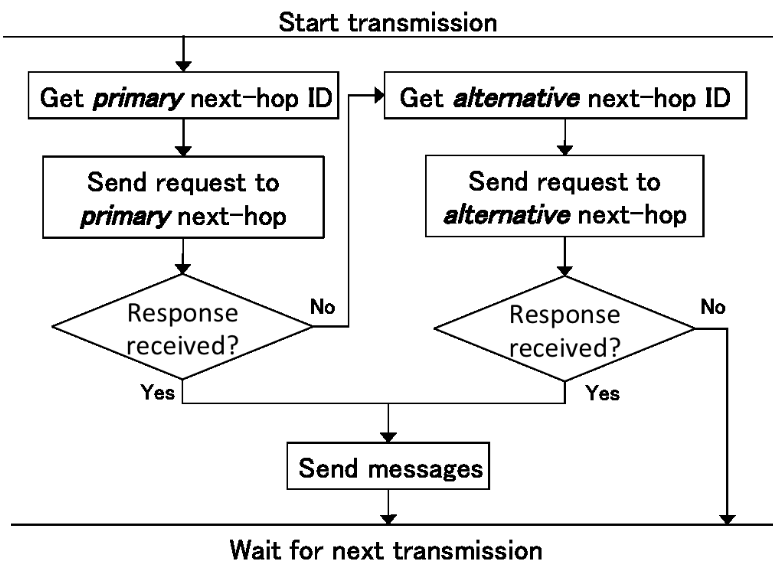


Figure 3.1: Overview of FAST.

3.2 Notations

Let N be a set of nodes in the network. A set of neighbour nodes of node $n \in N$ is denoted by $nbr(n)$. Each node holds a *potential* which is a positive scalar value representing the distance from the destination. Potential of node $n \in N$ for each destination $d \in N$ at time t is defined by $V^d(n,t)$. The potential starts from 0 at the destination ($V^d(d,t) = 0$) and increases linearly hop-by-hop from the destination.

Every node has next-hop information for each destination. We denote it by $NH^d(n,t)$, which shows the next-hop node for destination d of node n at time t.

Link quality and path quality are defined as *forward predictability* and *delivery predictability*. $P_F(n,k,t)$ denotes forward predictability of the link between node $n \in N$ and node $k \in nbr(n)$ at time t and $P_D^d(n,t)$ represents delivery predictability from node $n \in N$ to the destination $d \in N$ at timeslot t. Delivery predictability of the destination always ties to 1 ($P_D^d(d,t) = 1$) and decreases depending on the quality of links from the destination.

3.3 Unidirectional Link Detection

3.3.1 Definition

To classify links at each node $n \in N$, the link pointing outward from node n is defined as *forward link* and the link directing to node n is called *reverse link*. The definition is illustrated in the Figure 3.2.

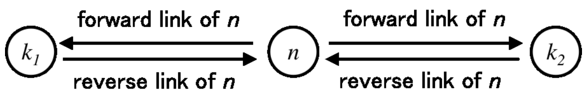


Figure 3.2: Link's definition.

We define that any link becomes unavailable when the link cannot deliver any messages within a period of time called *Connectivity Time* (T_C). Supposing that node n is connected with node $k \in nbr(n)$, when we consider at node n, there should be both forward link to node k and reverse link from node k if the link is bidirectional. The link between node n and node k becomes unidirectional when the forward link to node k is available but the reverse link from node k is unavailable or vice versa. The example of the unidirectional link is shown in Figure 3.3.

Here, T_C is the important factor to decide whether the link is unidirectional. Due to the fluctuation of wireless link, the messages could be lost, even though the link quality is good. T_C filters out ordinary message loss and confirms that the link is unidirectional.

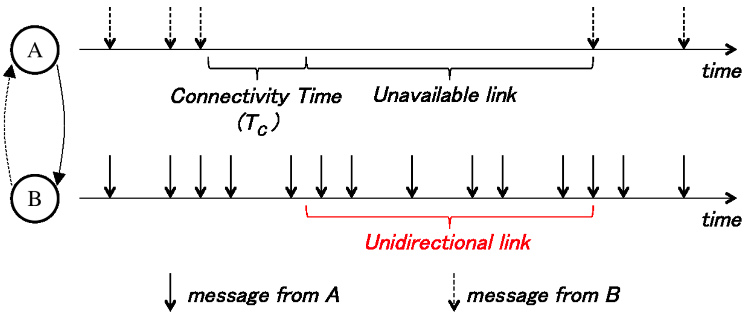


Figure 3.3: Example of the unidirectional link between node A and node B.

3.3.2 Method

To detect unidirectional links, a neighbour table at each node must consist of:

- Reverse Link Lifetime (RLL_k)
- Link status (L_k)

Reverse Link Lifetime (RLL_k) is non-negative integer showing time-to-live of the reverse link from neighbour k. Once RLL_k is set to non-zero value, RLL_k decreases continuously, for example, every second. Link status (L_k) indicates whether the link connected with neighbour k is bidirectional. L_k is either bidirectional ('B') or unidirectional ('U'); L_k is set to 'B' only when both reverse link and forward link are available and set to 'U' if either the reverse link or the forward link is unavailable.

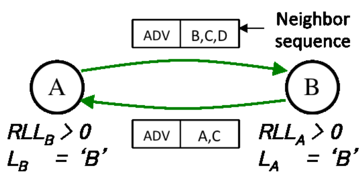
FAST detects unidirectional links between nodes by observing the receipt of ADVs on each link. Since each node broadcasts ADVs periodically, it learns that each reverse link is unavailable when it does not receive ADV within T_C. Whenever nodes receive an ADV from neighbours, nodes set RLL to T_C. Therefore, non-zero RLL indicates that the reverse link is available, while zero RLL means the reverse link is unavailable. Note that T_C must be more than ADV broadcast interval to detect the unidirectional links.

However, nodes cannot identify the presence of forward link by themselves. Nodes embed a neighbour sequence containing neighbour ID with non-zero RLL in ADV to inform other nodes. When nodes find themselves in the neighbour sequence, nodes perceive that the forward links are available. Then, nodes set L to 'B'. On the other hand, nodes realize that the links are unidirectional and set L to 'U' when their IDs are not in the neighbour sequence. An example of unidirectional link detection is illustrated in Figure 3.4.

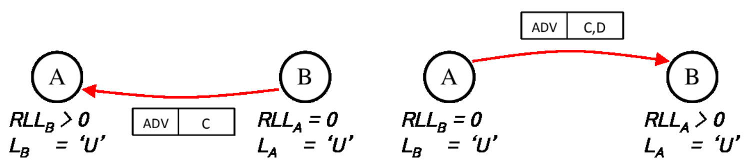
ADV reception and link information from neighbour nodes confirm the existence of reverse link and forward link respectively. Any node knows whether the forward link is available only if the reverse link exists. That is enough for DTN-based routing protocol because as long as the reverse link is lost, the node cannot update neighbours' information or succeed the investigation. Therefore, forward link information is unnecessary unless the reverse link is available.

3.4 Link Quality Estimation

Link quality metric helps FAST providing an alternative path for reinvestigation. In Ref. [21], the estimator measures the quality of each link by calculating the ratio of the number of receiving ADVs to the number of expected ADVs to find the delivery probability. FAST also estimates link by observing the receiving ADVs. However, FAST conducts investigation before sending data messages, therefore the estimator does not calculate the probability of delivery success directly. The estimator just predicts if a node can succeed the investigation.



(a) Bidirectional link



(b) Unidirectional link

Figure 3.4: Example of reverse link lifetime (RLL) and link status (L) when the link between node A and node B is (a) bidirectional and (b) unidirectional.

In this research, we establish link quality and path quality metric called *forward predictability (P_F)* and *delivery predictability (P_D)* respectively, where $P_F, P_D \in [0, 1]$. Forward predictability is the bidirectional link quality predicting if nodes can succeed and, as a result, have a chance to send the messages on each link. Delivery predictability is the multiplication of forward predictability forecasting the probability that a node is able to succeed the investigation along the path.

FAST simply estimates link quality by calculating Advertise Reception Ratio (ARR) as defined in Equation (2.3). We revisit the definition of ARR again here.

$$ARR_{k \to n} = \frac{N_{adv\_recv}(n, k)}{N_{adv\_send}(k)} \tag{3.1}$$

where $ARR_{k \to n}$ denotes ARR from node k to node n, $N_{adv\_recv}(n, k)$ is the number of ADVs from k received at n and $N_{adv\_send}(k)$ is the number of ADVs sent by k.

Theoretically, link quality should be calculated in a very short interval in order to get high accuracy. The most popular method is using reactive probe which generates lots of probe packets on the interested link. But this method consumes bandwidth and energy, so we decide to utilize ADVs instead. In practice, ADV broadcast interval is probably long for alleviating the congestion. Calculating ARR with a short period on the high-varied links will show the discrete change and instability. To smooth the discretion, we apply exponentially weighted moving average (EWMA) after computing ARR at each period as showing below:

$$\widehat{ARR}(t) = \alpha \times \widehat{ARR}(t - T_E) + (1 - \alpha) \times ARR \tag{3.2}$$

$\widehat{ARR}(t)$ denotes ARR after filtering by EWMA at time t, α is a smooth factor where $\alpha \in [0, 1]$ and T_E is the estimation period.

Then, forward predictability between node n and neighbour k and delivery predictability of node n for destination d are computed as follows,
Forward Predictability:

$$P_F(n, k, t) = \widehat{ARR}_{n \to k}(t) \times \widehat{ARR}_{k \to n}(t) \tag{3.3}$$

Delivery Predictability:

$$P_D^d(n, t) = \max_{k \in nbr(n)} \{P_D^d(k, t) \times P_F(n, k, t)\} \tag{3.4}$$

$$P_D^d(d, t) = 1 \tag{3.5}$$

Nodes share and update their link quality information with every neighbour by piggybacking those information with ADVs. Each node estimates ARR, as well as, $\widehat{ARR}$ of every link. Then, it adds its current P_D and $\widehat{ARR}$ in the neighbour sequence. Finally, ADV must contain information as shown in Figure 3.5. After nodes receive ADV, they search for their ID in the neighbour sequence and update P_F with received $\widehat{ARR}$ (filtered ARR of forward link) and local estimated $\widehat{ARR}$ (filtered ARR of reverse link). Then, nodes compute P_D and select the maximum multiple as its new P_D (Equation (3.4)).

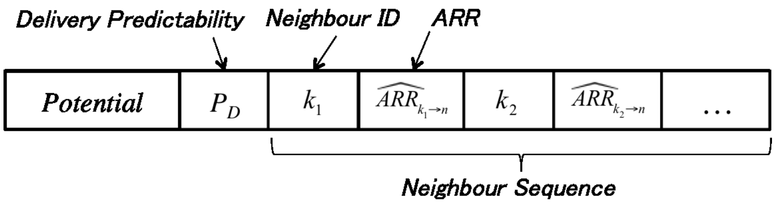


Figure 3.5: Information in advertisement message (ADV).

3.5 Next-hop Selection Scheme

FAST uses two next-hop selection scheme for primary next-hop and alternative next-hop. The primary next-hop is chosen based on the number of hops similar to PEAR. As we explained, PEAR's next-hop selection scheme chooses only the lowest potential node to be the next-hop without being aware of the existence of unidirectional links. In order to avoid this problem, FAST modifies PEAR's next-hop selection scheme and calls it *Unidirectional Link-Aware Next-hop Selection (ULANS).* ULANS is defined as follows,

$$F^d(n,t) = \max_{k \in nbr(n) \wedge L_k = \text{`}B\text{'}} \{V^d(n,t) - V^d(k,t)\} \quad (3.6)$$

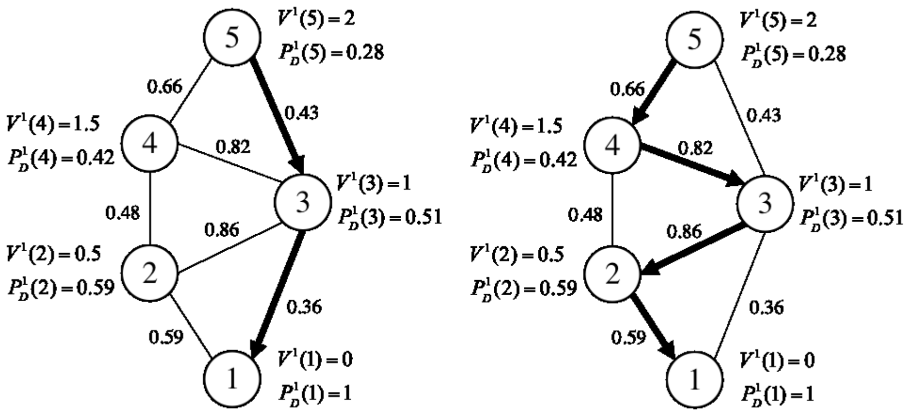
$$NH^d_{ULANS}(n,t) = \{k | k \in nbr(n) \wedge F^d(n,t) = V^d(n,t) - V^d(k,t)\} \quad (3.7)$$

ULANS adds one more condition to address the unidirectional links. When nodes compare the potential value with their neighbours, nodes consider only neighbours that bidirectional links exist (L_k = 'B'). As a result, ULANS forces nodes to choose only the lowest potential neighbour with bidirectional link as the next-hop.

The alternative next-hop is selected by considering the quality of links along the path from source to sink. FAST uses delivery predictability to decide the alternative next-hop as shown in Equation (3.8).

$$NH^d_{DP}(n,t) = \{k | k \in nbr(n) \wedge P^d_D(n,t) = P^d_D(k,t) \times P_F(n,k,t)\} \quad (3.8)$$

With this scheme, nodes select the neighbour that gives the highest delivery predictability among all neighbours for the reinvestigation. Figure 4.5 illustrates the example of paths established from primary next-hop and alternative next-hop in FAST. The number on a link represents forward predictability.



(a) Primary path (b) Alternative path

Figure 3.6: The example of primary path and alternative path from node 5 to node 1.

Chapter 4

Evaluation

This chapter presents the experimental evaluation of the proposed method. We carried out four experiments to evaluate the proposed forwarding scheme in various network sizes and conditions. First of all, we describe an overview of each experiment and give the definition of performance metrics we used in the evaluation. After that, we explain hardware and software's structure using in this experiment. Finally, the detail and results of four experiments are presented.

4.1 Methodology

FAST was evaluated by test bed experiment. We have implemented all the details of FAST described in the last chapter to PEAR and deployed it to UTMesh nodes. Since FAST inherits some mechanisms from PEAR, we made use of PEAR's original source code by modifying existing useful components and programmed FAST components additionally.

FAST comprises addtional three mechanisms which were developed in accordance with the problems mentioned in Section 2.3, i.e. unidirectional link detection, link quality estimator and alternative retransmission scheme. Each mechanism in FAST must be examined to verify that it can solve those problems efficiently in the interested network. Howbeit, one of the proposed solutions is a unidirectional link detection which identifies unidirectional links with connectivity time (T_C) of links. Different T_C values might affect the speed of detection and, as a result, influence routing performance. For this reason, the optimal T_C should be determined in advance to achieve the best performance.

Our retransmission scheme differs from the conventional schemes in that a sender retransmits lost messages to an alternative node. Further, FAST applied two routing metrics, i.e. hop and delivery predictability, to retransmission. Therefore, FAST should be proved that it can give better performance than the traditional retransmission schemes based on both hop and delivery predictability in aspect of delivery latency.

There are many factors that govern routing performance in wireless communication system as we addressed in Section 2.2. Some of them depend on physical deployment such as network size and network density, while some are application-specific e.g. message size. Indeed, these factors are particular and determined by practical use. We chose some possible values to demonstrate that our proposal works well in various conditions.

Accordingly, we did four experiments to evaluate FAST performance. In the first experiment, we observed the improvement of PEAR after applying ULANS with different T_C values, in addition to, studied feature and effect of unidirectional links. We selected the one that resulted the lowest delivery latency as the optimal T_C and used it in later experiments. The second experiment examined the performance of each mechanism of FAST comparing with PEAR in different size of networks. The third and last experiment evaluated FAST in various node densities and message sizes in comparison with traditional retransmission schemes.

The primary performance of interest in this research is delivery latency, however, we also examined other performance metrics to study how FAST affect to overall routing performance. Totally, we observed investigation success rate, delivery rate, delivery latency, hop count, copy count and buffer size. The definition of each metric is given as follows,

- Investigation Success Rate (ISR): the number of response messages received over the total number of request messages sent by the node.
- Delivery rate: the number of messages successively received by the destination over the total number of messages sent throughout the experiment.
- Delivery latency: the amount of time each message travels from the source to the destination.
- Hop count: the number of hops that each message travels from the source to the destination.
- Copy count: the number of copies of each message in the network during the delivery.
- Buffer size: the number of entries occupied in the buffer of each node.

The delivery latency is the main performance metric interested in this research. The routing scheme should give low delivery latency, while preserving high delivery rate. ISR is examined since the success of investigation indicates whether the messages are sent to the next-hop. Hop count and copy count implies the delivery pattern of the routing scheme, as well as, redundancy in the network. Buffer size is one of important metrics in DTN. Routing protocol should minimize utilized buffer to prevent buffer overflow. It also reflects on the congestion and resource constraint of nodes.

Overall performance of each metric is represented by the average value, except delivery latency. We use median and 99th percentile instead of average and maximum value respectively for delivery latency in order to avoid the outliers.

4.2 Implementation

4.2.1 Software

We programmed FAST as an extension of PEAR[1] in C. Figure 4.1 presents a block diagram of implemented software on any node $n \in N$ where $k \in nbr(n)$ and $d \in N$. Our system consists of six components. PEAR already introduced advertisement manager, potential table, next-hop table and message manager. We modified these components, except potential table, and added more components for the proposed scheme i.e. neighbour table and link quality estimator. Each component works as follows,

- **Advertisement Manager (AM):** AM broadcasts ADV periodically and receives ADV from neighbour nodes. For broadcasting, AM retrieves the latest information including $P_D(n,t)$ and $\{\widehat{ARR}_{k\rightarrow n} | RLL_k > 0\}$ from neighbour table (NT), in addition to, $V^d(n,t)$ from potential table (PT). Then, AM publishes these information to neighbours. After node n receives ADV, AM extracts $V^d(k,t)$, $P_D(k,t)$ and $\widehat{ARR}_{n\rightarrow k}$ from ADV and submits them to NT.

- **Potential Table (PT):** PT manages all neighbours' potentials ($V^d(k,t)$) and computes the potential of the next time slot ($V^d(k,t+1)$) according to Equation (A.1).

- **Neighbour Table (NT):** NT stores and provides all neighbours' information for AM, link quality estimator (LQE) and next-hop table (NHT). Each neighbour entry contains $RLL_k, L_k, N_{adv\_recv}(n,k)$, both

[1]http://sourceforge.net/projects/pear/files/

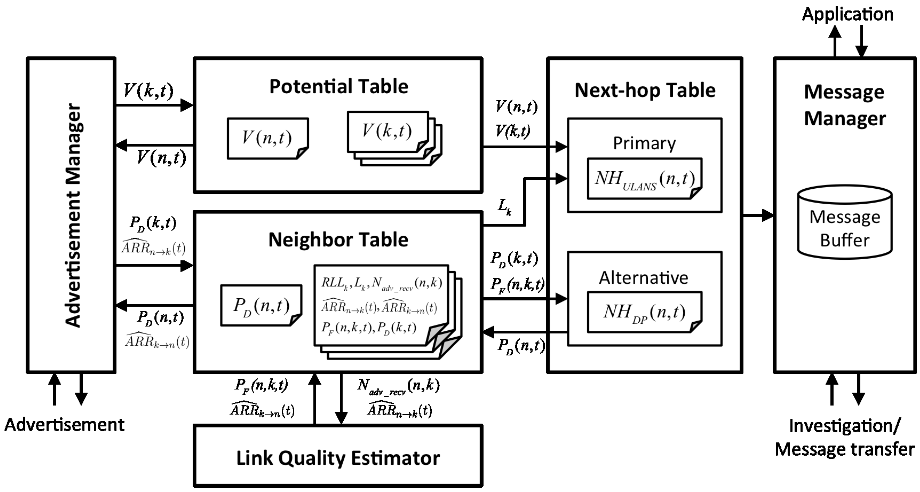


Figure 4.1: FAST system overview.

$\widehat{ARR}s, P_F(n,k,t)$ and $P_D(n,t)$. After receiving ADV from any neighbour k , NT sets $N_{adv\_recv}(n,k)$ to $N_{adv\_recv}(n,k)+1$ and RLL_k to connectivity time (T_C). It lessens RLL continuously, for instance, every second and updates L_k to 'U' when RLL reaches zero as explained in Section 3.3.

- **Link Quality Estimator (LQE):** LQE implements Equation (3.1) - (3.3) to measure link quality every estimation period (T_E). LQE obtains $N_{adv\_recv}(n,k)$ and $\widehat{ARR}_{n\to k}$ from NT, makes a computation and submits the updated $\widehat{ARR}_{k\to n}$ and $P_F(n,k,t)$ to NT.

- **Next-Hop Table (NHT):** NHT provides next-hop table for message manager (MM) using information from PT and NT. NHT consists of two modules: primary and alternative NHT. Primary NHT fetches $V^d(n,t)$ and $V^d(k,t)$ from PT along with L_k from NT. Then, primary NHT creates next-hop table based on ULANS (Equation (3.7)). Alternative NHT retrieves $P_D(k,t)$ and $P_F(n,k,t)$ from NT and generates next-hop table in accordance with Equation (3.8). After that, alternative NHT sends $P_D(n,t)$ back to NT for advertising this value to neighbours later.

- **Message Manager (MM):** MM provides application programming interface (API) for sending and receiving application messages. MM implements a replica and buffer management scheme discussed in Section A.0.5. In addition to PEAR's original code, we set up a investigation timer for retransmission. This timer is set to a short period

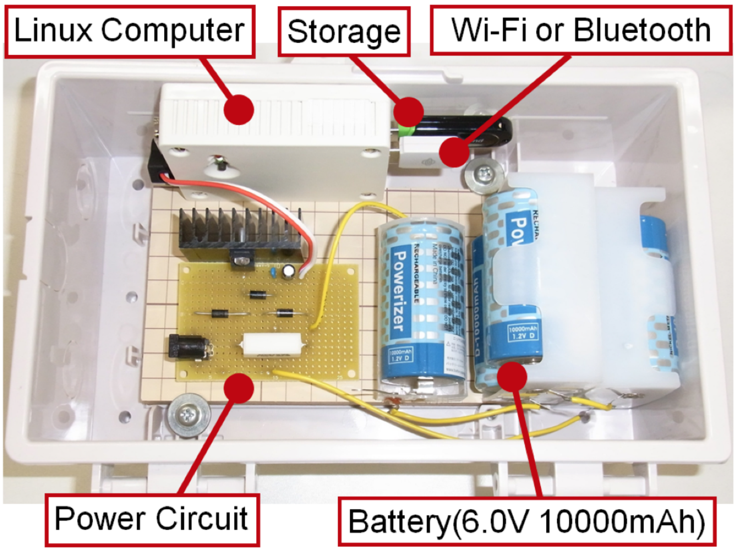


Figure 4.2: UTMesh node. [5]

called INV_TIMEOUT after a request message is transmitted to the primary next-hop. When the timer reaches zero, it triggers MM to send a request message again, but to alternative next-hop.

The original source code was already programmed to log useful routing information i.e. potential values, next-hop table, message buffer size, event of investigation and message transfer. Our additional modules record the detail in NT, e.g. $\widehat{ARR}, P_F$ and P_D, for examining network topology and link characteristics of the experimented networks.

4.2.2 Hardware

The implemented software was deployed to UTMesh [5] to evaluate our proposal. UTMesh is a testbed for wireless networking and DTN developed in the University of Tokyo.

The UTMesh node operates with Armadillo-200[2], an embedded computer with 8Mbyte program memory and 32MByte working memory. It works with ARM9 200MHz CPU and Linux operating system. USB WiFi (IEEE802.11) is added for ad-hoc communication. All working process is logged into the USB storage. The node can be powered by batteries or external power source. The components of UTMesh node is shown in Figure 4.2.

[2]http://www.atmark-techno.com/en/

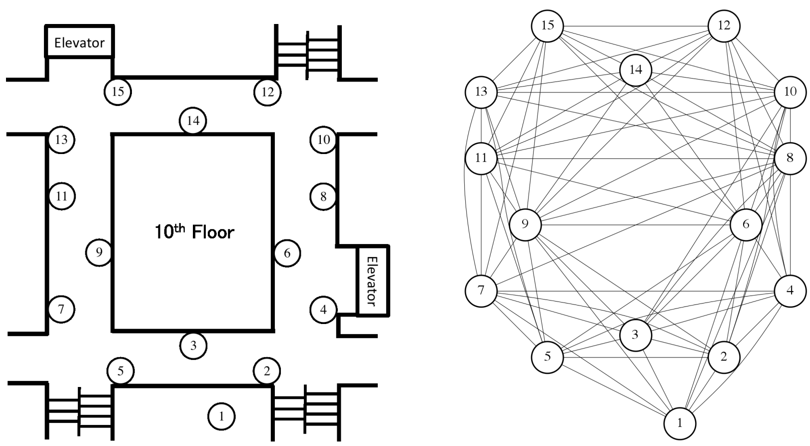
In our experiments, all the wireless interfaces were operated in ad hoc mode of 802.11b at the same frequency (2.412GHz) and same transmitted-power level. We charged all nodes at the same time before deploying them at each specified place on the floor. The working logs were retrieved for analysis after the experiment finished.

4.3 Experiment 1: Effect of connectivity time in ULANS

According to the definition of unidirectional links in this research, the connectivity time is an important variable to decide whether the link is unidirectional. We study the effect of connectivity time in this experiment.

4.3.1 Experiment setup

The experiment was carried out with 15 UTMesh nodes on 10^{th} floor of Eng. Bldg 2 in the University of Tokyo. We set system update interval (including ADV broadcast and next-hop selection interval) and retransmission interval to 10s. To mock-up monitoring scenario, every node was configured to send 22- or 23-byte messages (depends on node ID) to node 1 every 30s. Apart from PEAR, ULANS was run with 3 different T_C values, i.e. 40s, 60s, 80s. We denote the experiment of PEAR with ULANS as ULANS(T_C). Each experiment took 1 hour. Figure 4.3 shows the deployment, topology and



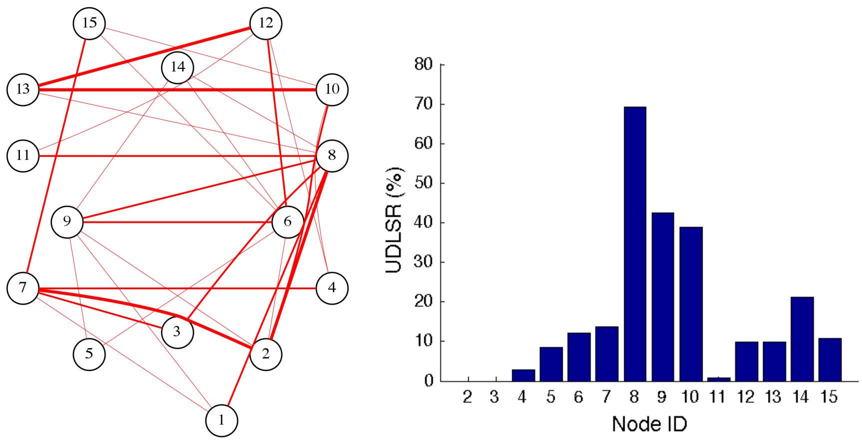
(a) Configuration (b) Topology

Figure 4.3: Deployment configuration and topology of the experimented network.

connectivity of this experiment.

4.3.2 Feature and effect of unidirectional links

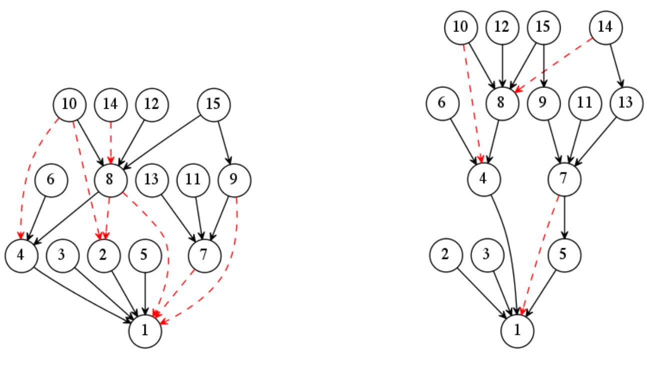
We picked PEAR and ULANS(80) to observe feature and effect of the unidirectional links to the delivery path. Figure 4.4a shows the presence of unidirectional links when T_C was 80s. The thicker line indicates that the link became unidirectional more frequently. From our observation, most links became unidirectional during the experiment, though all nodes oper-



(a) Unidirectional links existed in the network.

(b) Unidirectional link selection ratio of each node

Figure 4.4: Effect of unidirectional links when T_C was 80s.



(a) PEAR

(b) ULANS(80)

Figure 4.5: Main delivery paths of PEAR and ULANS(80). Dashed line indicates the unidirectional link between a pair of nodes.

ated at the same power level. However, their occurrence was dynamic. The links became unidirectional temporarily and turned back to be bidirectional back and forth. Some links became unidirectional only for a short time, while some links turned into unidirectional in a long period. Links between nodes located far from each other often became unidirectional.

We also observed how PEAR selected the next-hop by examining Unidirectional Link Selection Ratio (UDLSR) and main delivery paths. UDLSR is the number of times that nodes select next-hops with the unidirectional links over the number of times next-hop selection process is executed. Figure 4.4b shows UDLSR of PEAR when T_C was 80s. We can see that node 8, 9 and 10 gave high UDLSR which corresponded to the main delivery paths of PEAR as shown in Figure 4.5a. Those nodes often chose next-hops with the unidirectional links. After applying ULANS (Figure 4.5b), those nodes avoided the unidirectional links and chose next-hops with better links.

4.3.3 Results

We evaluated the improvement of PEAR with ULANS by observing ISR, delivery latency and hop count. We also compared maximum delivery latency in this experiment. Overall performance is presented in Table. 4.1.

Table 4.1: Overall performance.

Experiment	ISR(%)	Delivery Latency (s)			Hop count
		Median	99%	Max	
PEAR	56.22	10.12	205.22	401.60	1.78
ULANS(40)	66.03	20.01	228.60	382.54	2.33
ULANS(60)	63.08	9.15	136.85	297.00	2.09
ULANS(80)	63.12	9.18	135.56	201.68	2.21

ISR of the network was improved by 7-10% after applying ULANS to PEAR. In other words, avoiding the unidirectional links enhanced chances of sending the messages to the next-hop.

Figure 4.6 shows the distribution of delivery latency in a box plot. The whisker represents the 99^{th} percentile of delivery latency. Note that Figure 4.6 trims out the outliers of PEAR and ULANS(40). ULANS decreased maximum delivery latency, however, ULANS did not always reduce the median and 99^{th} delivery latency since ULANS(40) resulted higher value than PEAR. ULANS(60) and ULANS(80) outperformed PEAR. They improved the median delivery latency by 10%. Maximum delivery latency was decreased by 26% in ULANS(60) and 50% in ULANS(80). Moreover, 99% of the messages were delivered 33% faster in ULANS(60) and ULANS(80).

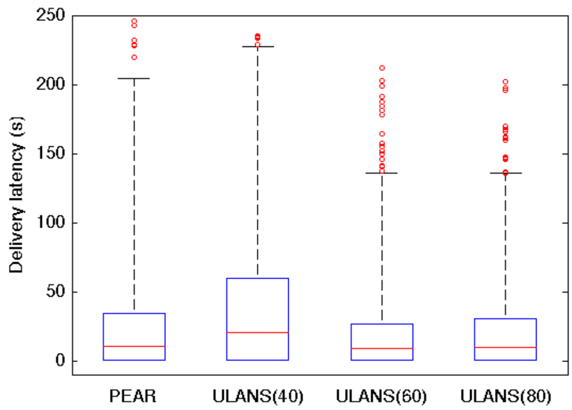


Figure 4.6: Box plot showing the distribution of delivery latency.

While PEAR chose the farthest nodes, which were likely to connect with the unidirectional links, as the next-hop, ULANS avoided those nodes and selected closer nodes resulting in more hop counts. The messages delivering with more hop counts possibly gave higher delivery latency. This is why median delivery latency was not always improved in ULANS.

From the results, the ratio of ADV broadcast interval and T_C that resulted the best performance on delivery latency is 1:6 (ULANS(60)). Therefore, we also used this ratio in the next experiments.

4.4 Experiment 2: Performance of proposed methods

This experiment studied the behaviour and improvement of each mechanism in FAST, including ULANS and delivery predictability-based next-hop selection, in comparison with PEAR. Effect of unidirectional links was studied again, but in the different deployment.

4.4.1 Experiment setup

We evaluated FAST forwarding scheme with two different deployments. UTMesh nodes were deployed on floor scenario and multistory scenario in Eng. Bldg. 2, The University of Tokyo. The detail and configuration of each deployment are shown in Table. 4.2 and Figure 4.7. In floor scenario,

Table 4.2: Detail of floor and multistory scenario.

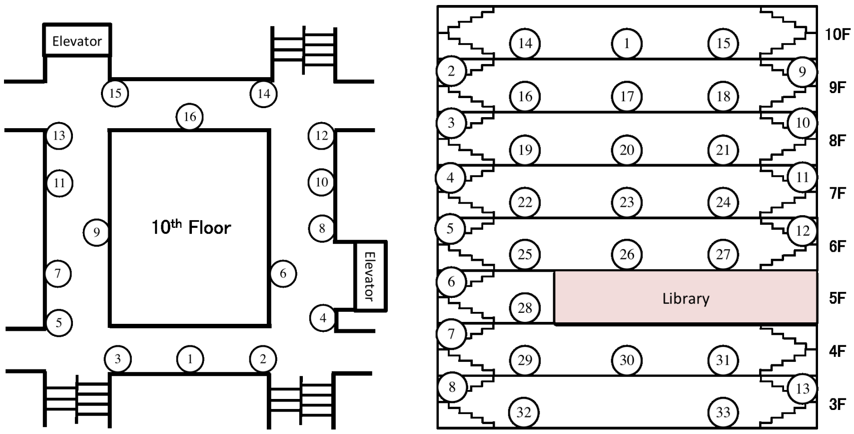
Detail	Floor	Multistory
Location (Floor)	10^{th}	3^{rd}-10^{th}
Number of nodes	16	33
Source Node ID.	2-16	14-33
Destination Node ID.	1	1
Experiment time/scheme	1 hr	1.5 hrs
Message size	22-23 bytes	
System update interval	5s	
Message generation interval	30s	
Retransmission interval	10s	
Connectivity time (T_C)	30s	
Link quality estimation interval (T_E)	30s	

all nodes except node 1 were the source nodes, but in multistory scenario, only nodes deployed on the corridor were the source nodes. Nodes on the stairs just relayed the messages from the sources to the destination.

Apart from PEAR, three more selection schemes were implemented in order to study the effect of unidirectional and low quality links on DTN-based routing protocol, in addition to, evaluate FAST by comparing with PEAR. In the experiment result, ULANS denotes the experiment with only ULANS (no retransmission). DP represents the experiment with delivery predictability-based next-hop selection scheme in Equation (3.8) (no retransmission). Finally, FAST is the experiment with the proposed scheme.

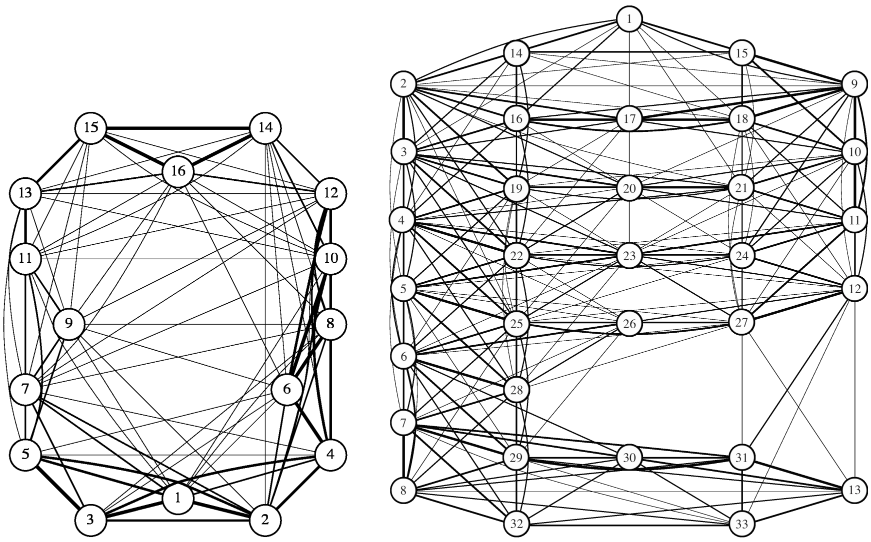
4.4.2 Feature of the networks

Figure 4.8 shows the topology and connectivity of floor and multistory scenario. A thickness of line implies the value of forward predictability of each link. The thicker line means higher forward predictability. Since our testbed was operated with WiFi, some nodes were connected across many floors in multistory scenario. Considering the links connecting between nodes on the same floor, short links mostly had higher forward predictability. However, the links connecting between nodes deployed on the different floor mainly had low forward predictability. Even though those links were shorter than



(a) Floor scenario (b) Multistory scenario

Figure 4.7: The deployment configuration of the experimented networks.



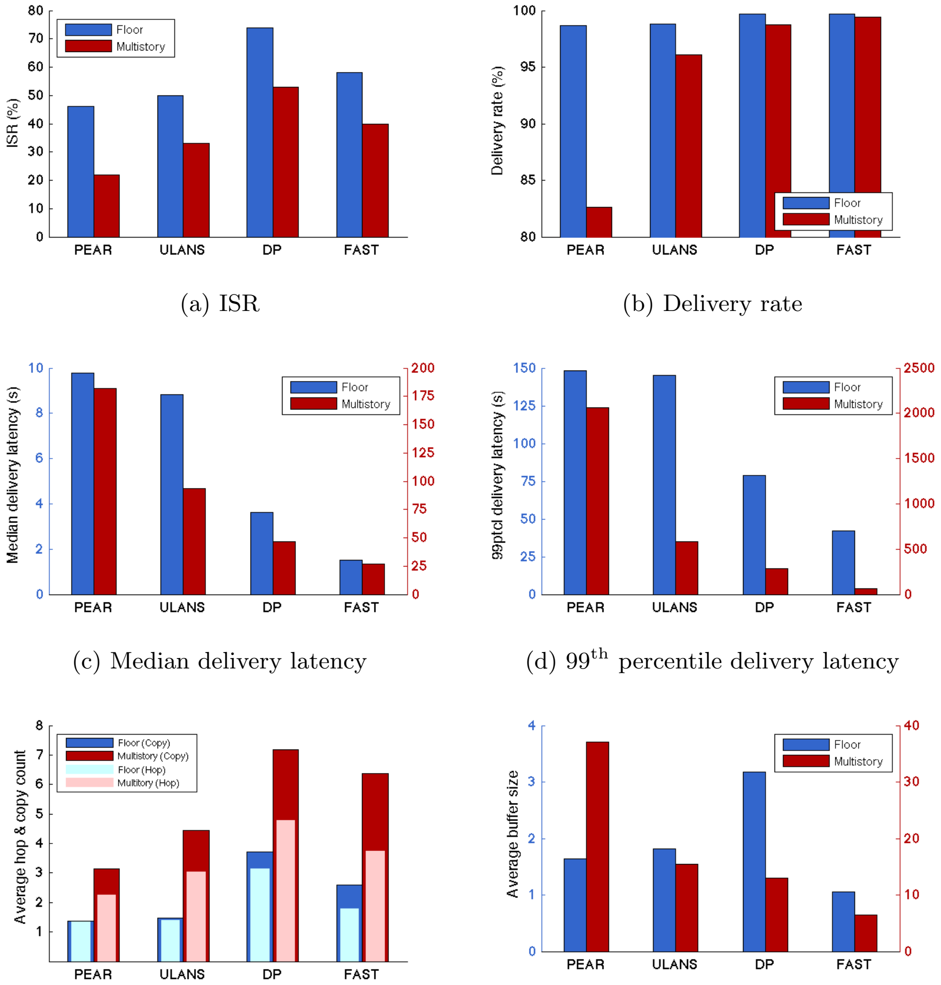
(a) Floor scenario (b) Multistory scenario

Figure 4.8: Topology and connectivity of experimented networks.

the links on the same floor, the signal strength was attenuated by the roof resulting in low forward predictability.

4.4.3 Results

All performance metrics were compared in this experiment. The overall performance is shown in Figure 4.9. We analyzed effect of unidirectional links, effect of low quality links and FAST performance as follows,



(a) ISR

(b) Delivery rate

(c) Median delivery latency

(d) 99$^{\text{th}}$ percentile delivery latency

(e) Average hop and copy count

(f) Average buffer size

Figure 4.9: Overall performance.

Effect of unidirectional links

Comparing the performance of PEAR and ULANS, the unidirectional links did not influence much on delivery rate and median delivery latency in floor scenario. On the other hand, the unidirectional links significantly reduced PEAR performance in multistory scenario.

We examined unidirectional link selection ratio (UDLSR) to prove that the unidirectional links were the cause of bad performance on delivery rate

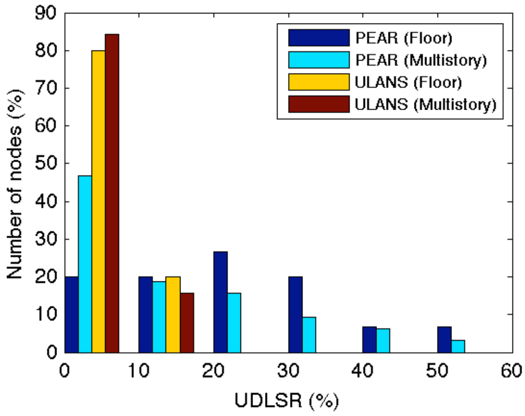


Figure 4.10: The distribution of unidirectional link selection ratio (UDLSR).

and delivery latency in PEAR. Figure 4.10 illustrates the distribution of UDLSR of PEAR and ULANS. We can see that some nodes frequently chose neighbours with unidirectional links in PEAR (40-60% of the experiment time), while all nodes avoided unidirectional links after applying ULANS and acheived better delivery latency and delivery rate.

Comparing with PEAR in multistory scenario, ULANS enhanced ISR approximately 10%. ULANS decreased 41.28% in the median and 76.43% in 99th percentile delivery latency, but ULANS increased average hop and copy count. The increment of hop count indicated that the unidirectional links were usually long links. Although the hop count was increased, when we compared hop count with median delivery latency as shown in Figure 4.11, the messages were delivered considerably faster, even though they were

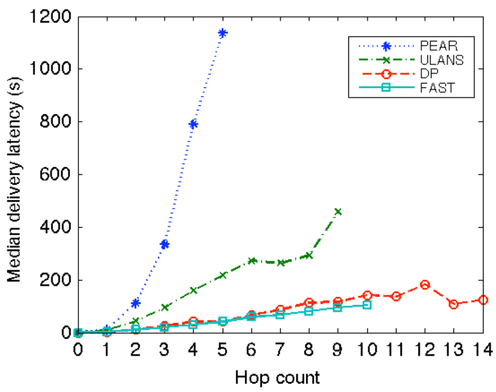


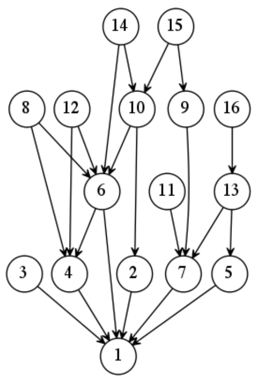
Figure 4.11: Hop count vs. Delivery latency in multistory scenario.

delivered through more hops.

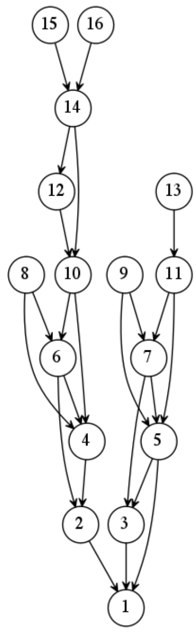
Average buffer size was increased in floor scenario. With short range links, ULANS created and sent more copied messages in the network, so the intermediate nodes received and stored more messages in their buffers. However, average buffer size was greatly decreased in multistory scenario. The reason was that PEAR deletes the messages that are informed as delivered messages. Seeing that delivery latency was greatly decreased in multistory scenario, the messages were quickly deleted from the buffer.

Effect of low quality links

Even though ULANS already filtered out unidirectional links, the rest of long links tended to be low quality links. DP selected high quality path, as well as, links which were particularly short links as shown in Figure 4.12. Obviously, the path from source to destination were longer, thus DP gave the most average number of hops and copies. The maximum hop count in DP were 8 hops in floor and 14 hops in multistory scenario. The results show that DP achieved the highest ISR. DP enhanced ISR around 20% compared to ULANS. Median and 99th percentile delivery latency were



(a) ULANS



(b) DP

Figure 4.12: Delivery path of ULANS and DP in floor scenario.

improved more than 50% in both scenarios. The improvement of delivery latency also benefited to delivery rate because some of messages were deleted from the network before reaching the destination due to message expiration. DP occupied more buffer entries in floor scenario, but it released buffer faster in multistory scenario similar to ULANS.

FAST Performance

With retransmission, FAST showed the best performance on delivery latency and delivery rate among all schemes, especially in multistory scenario. FAST had intermediate ISR since ISR included investigation to both primary and alternative next-hops. FAST reduced median delivery latency by 64% in floor scenario and 85% in multistory scenario in comparison with PEAR. Comparing with DP, median delivery latency was decreased by 59% in floor scenario and 43% in multistory scenario. 99% of the messages were gathered within one minute, while other schemes took more than 4 minutes in multistory scenario.

Certainly, FAST gave more hop count than PEAR, but lower than DP. The reason was that the messages were forwarded across some nodes when the sender succeeded in the investigation with the primary next-hop. However, FAST lessened few copy count since FAST created multi-path delivery. The messages were possibly forwarded to both primary and alternative next-hop at the different transmission time. For example, when the sender failed in the investigation with the primary next-hop but succeeded in the reinvestigation, the sender forwarded the messages to the alternative next-hop. At the next retransmission time, the sender was able to complete the investigation with the primary next-hop, so the sender sent the messages to the primary next-hop. In this case, the messages delivered to both primary and alternative next-hops. Such situation could happen at every intermediate node, thus FAST still had large copy count comparing to PEAR and ULANS.

FAST resulted the lowest average buffer size in both scenarios. The effect of delivery latency on the buffer size was more dominant than the copy count even in floor scenario.

According to our goal, the messages should be delivered to the destination within the sensing interval which was 30s in this experiment. Figure 4.13 illustrates the number of messages delivered to node 1 within 30s in percentage. FAST delivered almost all messages in floor scenario, while only half of transmitted messages reached the destination in multistory scenario. However, this number was enhanced by 30% compared to PEAR.

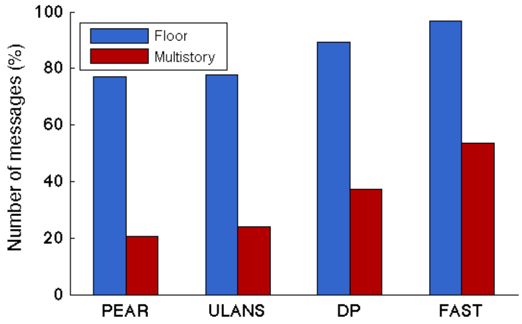


Figure 4.13: Number of messages delivered within 30s.

4.5 Experiment 3: Effect of node density

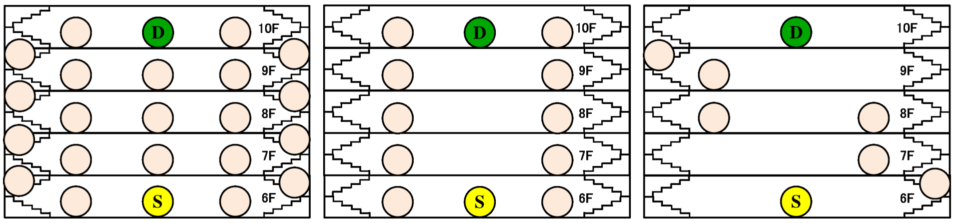
We have learned that FAST forwarding scheme outperformed PEAR in both small and large networks. In this experiment, we evaluate FAST in various node density and compare FAST with traditional retransmission schemes.

4.5.1 Experiment setup

To study the effect of node density, we use one-to-one communication instead of many-to-one communication. Only one source node periodically sent the messages to one destination. Location of source and destination was fixed, but the number of relay nodes was varied for each deployment. In this experiment, a node density is defined by an average node degrees ($\overline{deg}$) i.e. an average number of links connecting to a node.

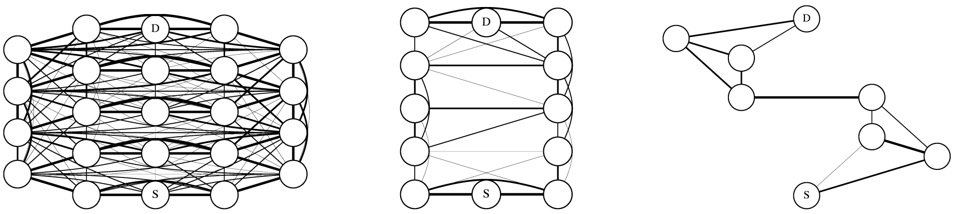
We evaluated the proposed scheme on three deployments as illustrated in Figure 4.14. The yellow node is the source and the green node is the destination. Topology and connectivity of each deployment is shown in Figure 4.15. Scenario A was a dense network consisting of 23 nodes with $\overline{deg} = 13$. Scenario B was sparser than Scenario A. It contained 12 nodes and had $\overline{deg} = 6$. Scenario C was the sparsest network where only 8 nodes were deployed with $\overline{deg} = 4$. This network was like two partitioned networks connected by only two nodes in the middle. All FAST parameters were similar to Experiment 2 in Table. 4.2. The experiment time was 1 hour for each scheme.

The proposed scheme was compared to the traditional retransmission schemes which retransmits lost messages to the same next-hop. Apart from FAST, two more schemes were implemented. The first one was hop-based retransmission. Each node retransmitted the request message only once to



(a) Scenario A (b) Scenario B (c) Scenario C

Figure 4.14: Configuration of the experimented networks.



(a) Scenario A (b) Scenario B (c) Scenario C

Figure 4.15: Topology of the experimented networks.

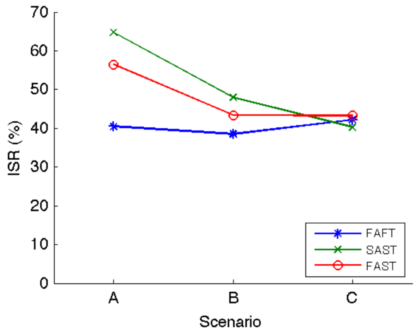
the same next-hop which was the farthest node in the transmission range. According to link characteristic, this scheme is denoted by FAFT (Farther-Aim-Farther-Try). FAFT next-hop was selected by ULANS to avoid effect of unidirectional links. The other scheme was link quality-based retransmission. The next-hop was selected based on link-quality, so this scheme is denoted by SAST (Shorter-Aim-Shorter-Try). SAST next-hop was decided based on delivery predictability-based next-hop selection.

4.5.2 Results

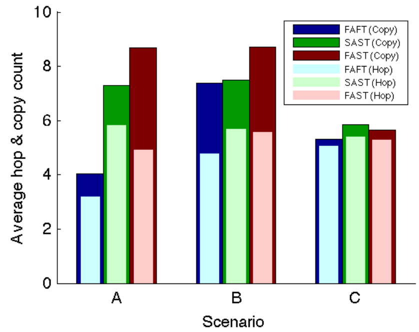
The performance metrics evaluated in this experiment were ISR, hop count, copy count and delivery latency. The overall performance is represented in Figure 4.16.

Thanks to investigation and retransmission, all schemes achieved more than 98% delivery rate in all scenarios. Besides, the number of entries occupied at each node was very small so that we cannot see the correlation of buffer size. Therefore, delivery rate and buffer size were not compared in this experiment.

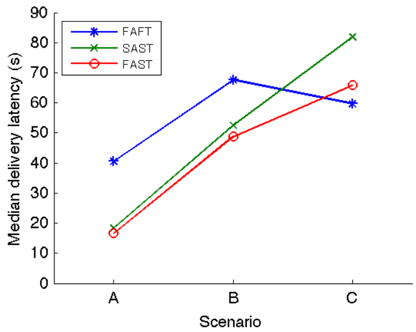
The trend of ISR and average hop count was similar to Experiment 2; FAFT resulted the lowest ISR and average hop count, while SAST gave the highest ISR and average hop count. FAST had both intermediate ISR and



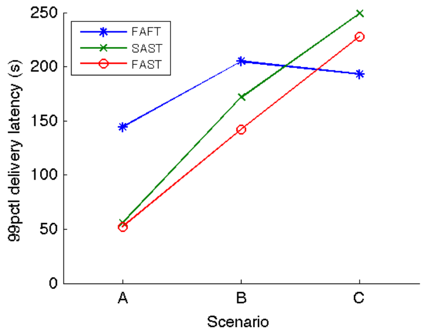
(a) ISR



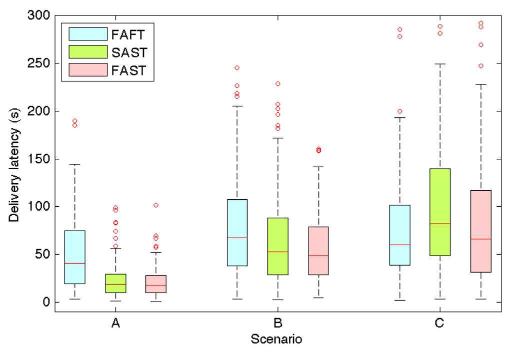
(b) Average hop and copy count



(c) Median delivery latency



(d) 99^{th} percentile delivery latency



(e) Distribution of delivery latency

Figure 4.16: Overall performance.

average hop count. The difference of ISR was obvious in dense network (Scenario A). Each node had several neighbours, so it had diverse links. SAST always chose better quality links, on the other hand, FAFT selected long links which caused high message drop. This difference became less when the network got sparser. In Scenario C, each node had only one or two neighbours for forwarding the message to the destination. Thus, ISR was not different much.

Delivery latency was higher in sparser networks. In Scenario A and B, FAFT resulted the worst performance, while FAST achieved the best performance on delivery latency. FAST gave little improvement compared to SAST as it reduced number of hops when it succeeded the investigation with the primary next-hop. In the sparsest network (Scenario C), it turned out that FAFT achieved the best performance and, in contrast, SAST resulted the worst performance on delivery latency. The reason was the topology of this network.

The example is shown in Figure 4.17. The number on the link represents forward predictability. There were 2 paths from node 1 to node 3. Based on path quality, node 1 chose node 2 as the next-hop. However, the link quality between node 2 and node 3 was as bad as the link quality between node 1 and node 3. In this case, forwarding messages directly to node 3 might be better. This situation caused high delivery latency in SAST resulting the worst delivery latency in the sparsest network. FAST was also affected from this problem, so FAST gave higher delivery latency than FAFT.

Average copy count was always more than average hop count because the dynamic of wireless link caused nodes to change next-hops and send another copy to another next-hop. However, FAST created the most copies for each message because each node could send messages to both primary and alternative next-hops meaning that there were multi-paths from source to destination. This is also the reason of lower delivery latency in FAST.

Figure 4.18 presents the percentage of messages delivered within generation interval which was 30s. FAST acquired data faster than the others. 80% of the message were delivered within 30s in the dense network, but this percentage decreased significantly in sparser network, only 30% delivered in

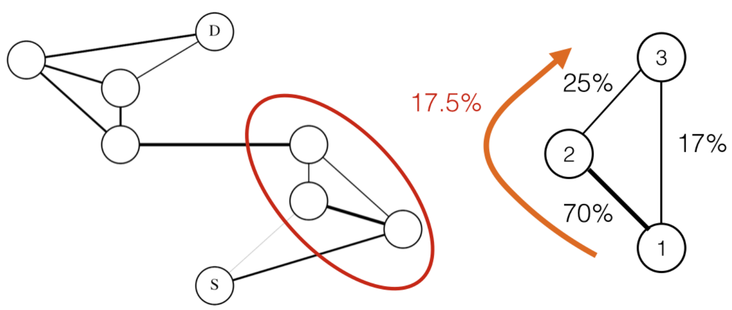


Figure 4.17: The example of problem in SAST.

scenario B. In scenario C, even though median delivery latency of FAST was higher than FAFT, this percentage was equal to FAFT's.

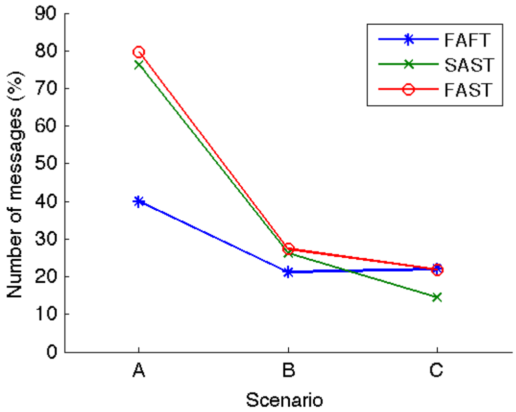


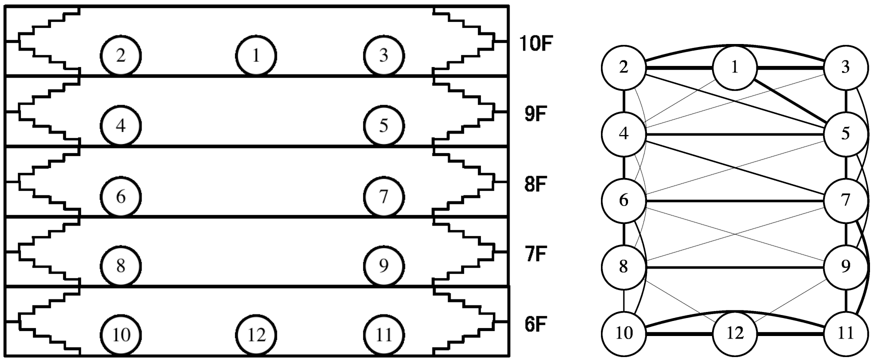
Figure 4.18: Number of messages delivered within 30s.

4.6 Experiment 4: Effect of message size

In reality, a sender assembles sensing data into one message or an intermediate node aggregates the receiving data together before forwarding the messages in order to reduce overhead, as well as, congestion in the network. Therefore, the message size could be various in WSNs. This experiment was carried out to study the effect of message size to FAST.

4.6.1 Experiment setup

Scenario B from Experiment 3 was used in this experiment, but we changed the communication to many-to-one communication. The experimented network consisted of 12 nodes where all nodes were the source node except node 1 which was the destination (Figure 4.19). FAST parameters were set similar to Experiment 2 (Table. 4.2) except retransmission interval. To lessen delivery latency, retransmission interval was changed to 5s (We used 10s in previous experiments). The experiment time was 1 hour for each scheme. Experimented message sizes were 100, 400, 700 and 1000 bytes. The proposed scheme was, again, compared to existing retransmission scheme: FAFT and SAST.



(a) Deployment (b) Topology

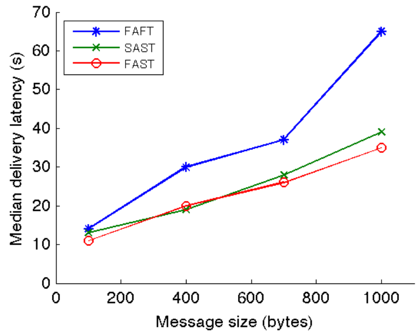
Figure 4.19: Configuration and topology of the experimented networks.

4.6.2 Results

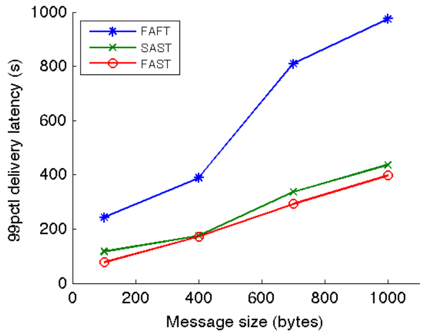
We observed delivery rate, delivery latency and buffer size in this experiment. With the same topology, ISR, hop and copy count should be constant. They was little varied due to dynamic links, but the variation did not effect much on these metrics. Overall performance is displayed in Figure 4.20.

In this case, delivery latency was mainly caused by the loss of data message rather than the failure of investigation. Because ISR and hop count was nearly constant, every node had a chance to send the data message equally in all message size. The larger the message size, the higher the message loss [41]. This, consequently, caused worse delivery latency and delivery rate. Nevertheless, FAST resulted lowest delivery latency and acquired data fastest for all message size. Median delivery latency was improved up to 10% and the number of message delivered within 30s was enhanced up to 5% in comparison with SAST.

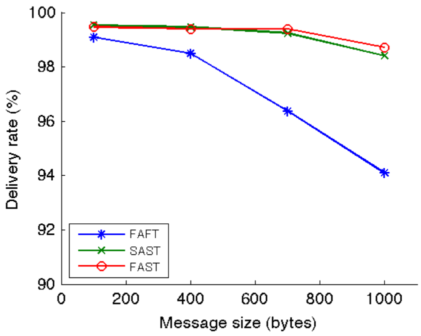
Delivery rate and buffer size were correlated with delivery latency. Delivery rate was also dropped when the message size increased. Although the investigation assured successful delivery, it was not enough for FAFT and large message size because delivery latency was too large so that some messages were expired and deleted from the network before reaching the destination. As larger messages were delivered to the destination slower, intermediates nodes stored messages longer resulting in larger buffer size. Nonetheless, FAST resulted the highest delivery rate and the lowest average buffer size since FAST achieved lowest delivery latency.



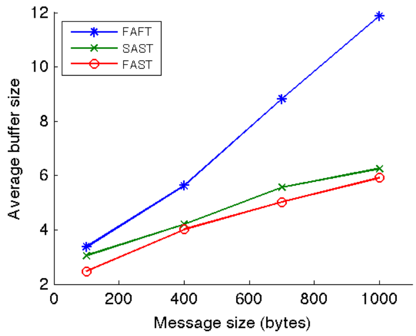
(a) Median delivery latency



(b) 99$^{\text{th}}$ percentile delivery latency



(c) Delivery rate



(d) Average buffer size

Figure 4.20: Overall performance.

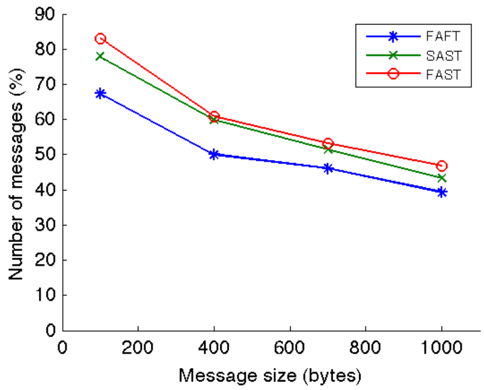


Figure 4.21: Number of messages delivered within 30s.

Chapter 5

Discussion

In this chapter, we discuss the performance of FAST and other factors that affected to FAST from the experimental results.

5.1 Performance of FAST

5.1.1 Comparing with PEAR

The results from Experiment 2 showed that the combination of DTN approach and WSN routing schemes in FAST improved PEAR's performance significantly on delivery latency and delivery rate. The improvement of delivery latency also benefited to the buffer size.

The previous work claimed that PEAR can achieve 100% delivery rate, but that work used one-to-one traffic pattern. Our experiment was carried out with more realistic many-to-one traffic pattern which generated more messages, as well as, congestion in the network causing more message loss. In large network like multistory scenario, the messages that generated by nodes locating far from destination took long time in the delivery. Owing to hop metric, some nodes failed in the investigation too many times so that the messages were discarded from the network due to the message expiration. This is the reason of imperfect delivery rate in our experiment.

The major problem in PEAR is the existence of unidirectional links. In accordance with UDLSR in Experiment 1 and Experiment 2, nodes often chose unidirectional links in next-hop selection. The unidirectional links led to high delivery latency in PEAR. ULANS enhanced successful investigation and, as a result, decreased delivery latency. ULANS also collected data faster in comparison with PEAR.

The results show that only hop metric and link quality metric is not enough to achieve satisfied delivery latency. DP decreased delivery latency more than 50% in both scenarios. Nevertheless, messages are likely lost any time owing to the dynamic of wireless links. Therefore, retransmission or retinvestigation in DTN approach is necessary for delivery latency improvement.

5.1.2 Comparing with traditional retransmission schemes

The results from Experiment 3 and Experiment 4 proved that FAST outperformed both FAFT and SAST on delivery latency in different conditions i.e. node densities and message sizes. Besides, FAST tended to achieve higher delivery rate compared to SAST in larger message sizes and denser networks.

In case of FAFT, retransmission did not help enhancing the performance much because nodes still suffered from repeatedly message loss on long links. In contrast, delivery latency was considerably decreased after changing to link quality-based scheme in SAST and FAST.

In most cases, FAST gave just little improvement (up to 10%) in comparison with SAST. This is because FAST delivers messages in the same way as SAST does if nodes fail in the first investigation every time. Seeing that investigation always failed on long links (primary next-hop), most messages are delivered on short links (alternative next-hop) similar to SAST.

Link quality metric does not perform well in all conditions. In sparse network (Scenario C in Experiment 3) where all links have almost equal quality, the results revealed that hop metric was better since link quality metric occasionally increased number of hops unnecessarily. Additionally, FAST's alternative next-hop is useful when the network is dense to some extent. Otherwise, the primary and alternative next-hop are the same nodes which equal to FAFT and SAST.

In this research, we implemented FAST to investigation. However, we believe that FAST can be applied to other forwarding schemes and give better performance on delivery rate, delivery latency and buffer size. For example, in classical ARQ, the sender might consider sending a message to alternative next-hop if it does not receive eACK.

5.2 Connectivity time in ULANS

As we mentioned in Section. 5.2, connectivity time (T_C) is the significant parameter to filter out the unidirectional links. T_C must be set corresponding to ADV broadcast interval. Certainly, we want to detect the unidirectional links as fast as possible, however, too small T_C is not enough for

ULANS to decide whether ADV loss is caused by unidirectional links or just normal packet loss. Nodes may change next-hops unnecessarily leading to higher hop counts and, consequently, higher average delivery latency as in ULANS(40) in Experiment 1. On the other hand, large T_C may lead to slow detection. In the case that the unidirectional links occur during the investigation process, ULANS have to wait for T_C to acknowledge this occurrence. Nodes still tried sending the messages through the unidirectional links uselessly. Seeing that different T_C gives different performance, T_C must be set appropriately.

5.3 Deployment and network scale

The performance of routing scheme depends on node deployment and network scale. We can see that floor and multistory scenarios in Experiment 2 gave different results in some performance metrics. The unidirectional links did not influence much in floor scenario, thus all schemes achieve more than 99% delivery rate. In addition, even though ULANS and DP delivered messages faster than PEAR, the average buffer was higher in floor scenario. Hence, other configurations possibly give different performance.

Deployment also has an effect on node density. We tried to carry out experiments in many configurations to study the behaviour of FAST and confirm that our FAST can work well in practice. However, Scenario C in Experiment 3 is a rare case and not recommended. We may need to cut the cost of deployment by reducing the number of nodes in reality, but as we have seen, too sparse network degrades the performance of routing protocols.

5.4 Redundancy

There is definitely a trade-off in routing scheme. While PEAR minimizes hop and copy count, next-hop selection schemes in FAST (i.e. ULANS and delivery predictability-based next-hop selection) increases the number of hops as well as copied messages. Besides, reinvestigation with alternative next-hop also creates multi-path from source to destination. This redundancy was apparent in Experiment 3 where there was only one source in the network. Copy count could be a bit more than hop count due to the fluctuation of wireless links in practice. However, according to results from Experiment 3, average copy count was 1.8 times more than average hop count in FAST. Furthermore, redundancy may cause bottleneck around nodes located near sink in our configurations which influenced higher message drop. This is one of the causes why delivery latency in FAST was not improved much compared with SAST.

Redundancy not only leads to congestion, but also affect to energy efficiency of sensor node which is the important issue of WSN. Even though we did not take energy efficiency and congestion into account much, future works should reduce this redundancy for more practicability.

Chapter 6

Conclusion

In this thesis, we addressed the issues concerning delivery latency of DTN-based sensor network in monitoring applications. DTN routing was originally relied on node mobility, however, nodes in monitoring applications are stationary. Thus, DTN approach is sometimes not suitable for this kind of network. We found that the major cause of delivery latency was repeatedly message loss on unreliable and unpredictable wireless links. DTN mainly selects the minimum-hop path containing long links, which frequently become unidirectional and have low quality.

We proposed FAST forwarding scheme to solve these issues in DTN-based sensor network. FAST was designed by taking many considerations into account. FAST inherits potential-based routing protocol (PBR) and the investigation from PEAR. PBR is energy efficient, fault tolerant and adaptive routing protocol which is famous for monitoring applications, while the investigation has shown high reliability and scalability in real-word experiment. FAST integrates retransmission and link quality metric from WSN routing schemes with hop-by-hop delivery in DTN. FAST relieves the effect of unidirectional links by Unidirectional Link-Aware Next-hop Selection (ULANS) scheme and enhances investigation success with delivery predictability-based next-hop selection in the reinvestigation.

Our experimental evaluation proved that FAST can considerably decrease delivery latency, while achieving high reliability and scalability on both small and large scale of networks in comparison with PEAR. Moreover, the results showed that retransmission with alternative next-hop alleviated delivery latency in diverse network densities and message sizes. Besides, the improvement of delivery latency also gave advantages to delivery rate and buffer size.

However, FAST relies on multi-path which generates more redundancies, i.e. hops and copies, in the network. Redundancy degrades WSN in terms of energy efficiency which is one of the major challenges in WSN. In additional, there are still many considerations we should concern. For example, monitoring systems possibly consists of various kinds of devices from different vendors. Therefore, these devices may have different hardware restriction and data. As a result, we should include heterogeneity in routing design. For more practicability, the future works should consider on energy efficiency and heterogenous network in routing design.

Appendix A

Potential-based Entropy Adaptive Routing (PEAR)

A.0.1 Notations

Let N be a set of nodes in the network. neighbour node of node $n \in N$ is denoted by $nbr(n)$. Potential of $n \in N$ for each destination $d \in N$ at time t is defined by $V^d(n,t)$. The potential at the destination always ties to zero i.e. $V^d(d,t) = 0$.

A.0.2 Potential field construction

PEAR nodes autonomously develop potential values by broadcasting their potential vector. Each node computes potential values in the following rule:

$$V^d(n, t+1) = V^d(n,t) + D \min_{k \in nbr(n)} \left\{ V^d(k,t) - V^d(n,t) \right\} + \rho \qquad \text{(A.1)}$$

The potential of destination is always tied to 0 ($V^d(d,t) = 0$). The potential increases by ρ at every period, but decreases when a node comes across another node holding smaller potential value. D is a diffusion parameter indicating agility of potential adaptation. Note that $0 < \rho < D$ and $0 < D < 1$. If the network is stable, Equation (A.1) turns into

$$\lim_{t \to +\infty} V^d(n,t) = \frac{\rho}{D} h(d,n) \qquad \text{(A.2)}$$

where $h(d,n)$ is the minimum hops from destination d to node n. A potential field converges into the same pattern that distance-vector routing i.e. the potential increases linearly hop-by-hop from the destination. In reality, as wireless links are intermittently-connected, the potential values vary all the time as illustrated in Figure A.1.

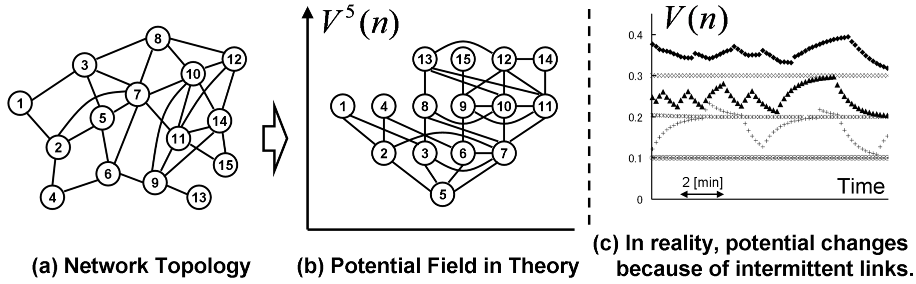


Figure A.1: An example of potential field construction.

A.0.3 Forwarding scheme

PEAR's next-hop selection scheme is described as follows,

$$F^d_{max}(n,t) = \max_{k \in nbr(n)} \{V^d(n,t) - V^d(k,t)\} \quad \text{(A.3)}$$

$$NH^d(n,t) = \{k | k \in nbr(n) \wedge F^d_{max}(n,t) = V^d(n,t) - V^d(k,t)\} \quad \text{(A.4)}$$

where $F^d_{max}(n,t)$ is the maximum difference of node n's potential and its neighbours' potential and $NH^d(n,t)$ is the next-hop of node n for the destination d at time t.

Firstly, nodes compare their potential with neighbours (Equation (A.3)). Then, the neighbours that give the maximum potential difference are chosen to be the next-hops (Equation (A.4)). Since the potential indicates the distance from the destination, the next-hop is the neighbour located farthest toward the destination in the transmission range.

A.0.4 Investigation

PEAR derives hop-by-hop reliable transfer with store-and-forward mechanism from DTN and modifies exchange protocol proposed by epidemic routing. Every node copies and stores receiving messages in the buffers. Before forwarding the messages, nodes investigate if the messages have already been sent to the next-hop or delivered to the destination. PEAR calls this process as *Investigation*. A sender sends a request message containing message IDs stored in its buffer to its next-hop. Then, the next-hop replies a response message consisting of the state of each message i.e. already-received, not-received or delivered. After that, the sender sends only not-received messages to the next-hop. In fact, PEAR does not only guarantee hop-by-hop delivery, but also assure end-to-end delivery implicitly. PEAR does not apply retransmission in forwarding. Nodes periodically investigate and send the messages in the buffers instead.

A.0.5 Replica and buffer management

Replica and buffer management is carried out to reduce the overhead of message duplication and to efficiently utilize buffer . In PEAR, each message must contain the following information:

- Message ID
- Destination ID
- Time-to-live (TTL)

Additionally, every node should have IsDelivered entry for each message in the buffer.

MessageID must be unique in the network. TTL is a message life time which decreases continuously (e.g. every second). When TTL reaches zero, the message expires and buffer entry including header of that message is freed. TTL corresponds to the left time for delivery deadline. IsDelivered shows whether delivery has been certified or not. When nodes receive a message, IsDelivered is set to false. After finding the delivery certification, IsDelivered is set to true and the body of message is deleted from the buffer. The first certificate will be published by the destination node after it has received the message. Note that the header of message is still preserved in the buffer after the node is informed the certificate in order to prevent redundant duplication. By this way, the message body is evicted from the buffer when it is informed as delivered message or when TTL reaches zero.

This management associates with investigation. When a node receives the request message, it checks status of each message in its buffer and replies one of

- MESSAGE_NOT_HAVE
- MESSAGE_ALREADY_HAVE
- MESSAGE_DELIVERED

In case of no header, it replies with MESSAGE_NOT_HAVE status. If the header exists, but IsDelivered is false, it responses with MESSAGE_ALREADY _HAVE status. Otherwise, it answers with MESSAGE_DELIVERED status. The sender will transmit only message with MESSAGE_NOT_HAVE status and delete body of message with MESSAGE_DELIVERED status.

Research activities

- Manussanun Buranachokphaisan, Hideya Ochiai, Hiroshi Esaki, "Unidirectional Link-Aware DTN-based Sennsor Network in Building Monitoring Scenario", Poster Session, Wireless days 2014, Rio de Janeiro, Brazil, November 2014.

- Manussanun Buranachokphaisan, Hideya Ochiai, Hiroshi Esaki, "Message Delivery Latency Improvement in DTN-based Sensor Network using Path Reliability Metric", 7th Thailand-Japan International Academic (TJIA) Conference 2014, Tokyo, November 2014.

- Manussanun Buranachokphaisan, Hideya Ochiai, Hiroshi Esaki, "FAST-DTN: Farther-Aim-Shorter-Try Disruption Tolerant Network for Building Monitoring Applications", 2014 IEEE Conference on Wireless Sensors (ICWiSe 2014), Kuala Lumpur, October 2014.

- Manussanun Buranachokphaisan, Hideya Ochiai, Hiroshi Esaki, "Disruption Tolerant Wireless Sensor Network for Building Energy Management System", Poster Session, Internet Conference 2013, pp. 137-138, Tokyo, October 2013.

Bibliography

[1] A. Tiab and L. Bouallouche-Medjkoune, "Routing in industrial wireless sensor networks: A survey," *Chinese Journal of Engineering*, vol. 2014, 2014.

[2] D. of energy, "Communications requirement of smart grid technologies," 2010.

[3] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 27–34, ACM, 2003.

[4] H. Ochiai, M. Nakayama, and H. Esaki, "Hop-by-hop reliable, parallel message propagation for intermittently-connected mesh networks," in *Proceedings of the 2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, WOWMOM '11, (Washington, DC, USA), pp. 1–9, IEEE Computer Society, 2011.

[5] H. Ochiai, K. Matsuo, S. Matsuura, and H. Esaki, "A case study of utmesh: Design and impact of real world experiments with wi-fi and bluetooth devices," in *Applications and the Internet (SAINT), 2011 IEEE/IPSJ 11th International Symposium on*, pp. 433–438, IEEE, 2011.

[6] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, pp. 14–27, ACM, 2003.

[7] K. Fall, W. Hong, and S. Madden, "Custody transfer for reliable delivery in delay tolerant networks," *IRB-TR-03-030, July*, 2003.

[8] A. Vahdat, D. Becker, *et al.*, "Epidemic routing for partially connected ad hoc networks," tech. rep., Technical Report CS-200006, Duke University, 2000.

[9] H. Ochiai and H. Esaki, "Mobility entropy and message routing in community-structured delay tolerant networks," in *Proceedings of the 4th Asian Conference on Internet Engineering*, AINTEC '08, (New York, NY, USA), pp. 93–102, ACM, 2008.

[10] J. Al-Karaki and A. Kamal, "Routing techniques in wireless sensor networks: a survey," *Wireless Communications, IEEE*, vol. 11, no. 6, pp. 6–28, 2004.

[11] R. Draves, J. Padhye, and B. Zill, "Comparison of routing metrics for static multi-hop wireless networks," in *ACM SIGCOMM Computer Communication Review*, vol. 34, pp. 133–144, ACM, 2004.

[12] D. S. De Couto, D. Aguayo, B. A. Chambers, and R. Morris, "Performance of multihop wireless networks: Shortest path is not enough," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 83–88, 2003.

[13] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325 – 349, 2005.

[14] A. Basu, A. Lin, and S. Ramanathan, "Routing using potentials: A dynamic traffic-aware routing algorithm," in *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '03, (New York, NY, USA), pp. 37–48, ACM, 2003.

[15] C. Schurgers and M. B. Srivastava, "Energy efficient routing in wireless sensor networks," in *Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force. IEEE*, vol. 1, pp. 357–361, IEEE, 2001.

[16] J. Wu, M. Lu, and F. Li, "Utility-based opportunistic routing in multi-hop wireless networks," in *Distributed Computing Systems, 2008. ICDCS'08. The 28th International Conference on*, pp. 470–477, IEEE, 2008.

[17] F. Khadar and T. Razafindralambo, "Performance evaluation of gradient routing strategies for wireless sensor networks," in *NETWORKING 2009*, pp. 535–547, Springer, 2009.

[18] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, "A multi-radio unification protocol for ieee 802.11 wireless networks," in *Broadband Networks, 2004. BroadNets 2004. Proceedings. First International Conference on*, pp. 344–354, IEEE, 2004.

[19] G. Parissidis, M. Karaliopoulos, R. Baumann, T. Spyropoulos, and B. Plattner, "Routing metrics for wireless mesh networks," in *Guide to Wireless Mesh Networks*, pp. 199–230, Springer, 2009.

[20] D. S. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," *Wireless Networks*, vol. 11, no. 4, pp. 419–434, 2005.

[21] M. D. Yarvis, W. S. Conner, L. Krishnamurthy, J. Chhabra, B. Elliott, and A. Mainwaring, "Real-world experiences with an interactive ad hoc sensor network," in *Parallel Processing Workshops, 2002. Proceedings. International Conference on*, pp. 143–151, IEEE, 2002.

[22] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE mobile computing and communications review*, vol. 7, no. 3, pp. 19–20, 2003.

[23] R. Ramanathan, R. Hansen, P. Basu, R. Rosales-Hain, and R. Krishnan, "Prioritized epidemic routing for opportunistic networks," in *Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking*, pp. 62–66, ACM, 2007.

[24] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '04, (New York, NY, USA), pp. 145–158, ACM, 2004.

[25] E. P. Jones, L. Li, J. K. Schmidtke, and P. A. Ward, "Practical routing in delay-tolerant networks," *Mobile Computing, IEEE Transactions on*, vol. 6, no. 8, pp. 943–959, 2007.

[26] Y. Feng, M. Liu, X. Wang, and H. Gong, "Minimum expected delay-based routing protocol (medr) for delay tolerant mobile sensor networks," *Sensors*, vol. 10, no. 9, pp. 8348–8362, 2010.

[27] M. Park, S. Min, S. So, D. Oh, B. Kim, and J. Lee, "Mobility pattern based routing algorithm for delay/disruption tolerant networks," in *Smart Spaces and Next Generation Wired/Wireless Networking* (S. Balandin, R. Dunaytsev, and Y. Koucheryavy, eds.), vol. 6294 of *Lecture*

Notes in Computer Science, pp. 275–286, Springer Berlin Heidelberg, 2010.

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

[29] N. Reijers, G. Halkes, and K. Langendoen, "Link layer measurements in sensor networks," in *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on*, pp. 224–234, IEEE, 2004.

[30] A. Cerpa, J. L. Wong, M. Potkonjak, and D. Estrin, "Temporal properties of low power wireless links: modeling and implications on multi-hop routing," in *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pp. 414–425, ACM, 2005.

[31] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, "An empirical study of epidemic algorithms in large scale multihop wireless networks," tech. rep., 2002.

[32] J.-B. Lee, Y.-B. Ko, and S.-J. Lee, "Euda: Detecting and avoiding unidirectional links in ad hoc networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 8, pp. 63–67, Oct. 2004.

[33] M. Zuhairi, H. Zafar, and D. Harle, "On-demand routing with unidirectional link using path loss estimation technique," in *Wireless Telecommunications Symposium (WTS), 2012*, pp. 1–7, April 2012.

[34] C. Perkins, E. Belding-Royer, S. Das, *et al.*, "Rfc 3561-ad hoc on-demand distance vector (aodv) routing," *Internet RFCs*, pp. 1–38, 2003.

[35] L. Sang, A. Arora, and H. Zhang, "On exploiting asymmetric wireless links via one-way estimation," in *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '07, (New York, NY, USA), pp. 11–21, ACM, 2007.

[36] B. Chen, S. Hao, M. Zhang, M. C. Chan, and A. Ananda, "Deal: Discover and exploit asymmetric links in dense wireless sensor networks," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON '09. 6th Annual IEEE Communications Society Conference on*, pp. 1–9, June 2009.

[37] H. She, Z. Lu, A. Jantsch, D. Zhou, and L.-R. Zheng, "Analytical evaluation of retransmission schemes in wireless sensor networks," in *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*, pp. 1–5, April 2009.

[38] M. A. Mahmood and W. K. Seah, *Reliability in wireless sensor networks: Survey and challenges ahead.* School of Engineering and Computer Science, Victoria University of Wellington, 2012.

[39] R. Gonzalez and M. Acosta, "Evaluating the impact of acknowledgment strategies on message delivery rate in wireless sensor networks," in *Communications (LATINCOM), 2010 IEEE Latin-American Conference on*, pp. 1–6, Sept 2010.

[40] C. Wang, K. Sohraby, B. Li, M. Daneshmand, and Y. Hu, "A survey of transport protocols for wireless sensor networks," *Ieee Network*, vol. 20, no. 3, pp. 34–40, 2006.

[41] J. Korhonen and Y. Wang, "Effect of packet size on loss rate and delay in wireless links," in *Wireless Communications and Networking Conference, 2005 IEEE*, vol. 3, pp. 1608–1613, IEEE, 2005.