University of St Andrews

CS4099

ILNP Routing for IoT

Author:
Jordan Mackie

Supervisor:
Prof Saleem Bhatti

April 12, 2019

Abstract

This project provides an implementation of a wireless ad-hoc sensor network, using ILNP as an addressing scheme. It focuses on an agricultural scenario, where groups of sensors are able to monitor and manage their local environment, whilst also providing data to a sink node for more in depth analysis. A zoning approach was taken to route data through the WSN, where subnetworks were created within the WSN in order to provide energy effecient routing with low memory and communication overhead. Finally, an experiment is carried out to show how effective the protocol is when using ILNP and IP as an addressing scheme, and the effectiveness of the protocol overall. The results showed that a protocol utilizing the locators-identifier split in ILNP can reduce the amount of network knowledge required for effective communication dramatically.

Declaration

I declare that the material submitted for assessment is my own work except where credit is explicitly given to others by citation or acknowledgement. This work was performed during the current academic year except where otherwise stated. The main text of this project report is 7326 words long, including project specification and plan. In submitting this project report to the University of St Andrews, I give permission for it to be made available for use in accordance with the regulations of the University Library. I also give permission for the title and abstract to be published and for copies of the report to be made and supplied at cost to any bona fide library or research worker, and to be made available on the World Wide Web. I retain the copyright in this work.

Contents

1 Introduction **1**

2 Context Survey **7**

3 Protocol Design **11**
3.1 Motivation and Overview 11
3.2 Intra-Zone Routing Protocol 12
3.3 Inter-Zone Routing Protocol 16

4 ILNP Testbed Implementation **20**
4.1 Wireless Network Emulation 20
4.2 Routing and Network Management 22
4.3 Sensor Emulation 23

5 Experiment **24**
5.1 Setup 25
5.2 Method 26
5.3 Results 26

6 Conclusion **31**

7 Appendix **33**

1 Introduction

As technology is becoming smaller and cheaper, more and more everyday objects are being connected to the internet. These devices form the Internet of Things (IoT). Examples of IoT devices range from smart televisions to simple heat sensors. By adopting the standard methods of communication used by the internet, these devices are able to communicate both with computers and servers, and amongst themselves.

The latter scenario is common for environments where the devices are spread over a greater distance than a wireless router would be able to serve them all directly. When this is the case, the devices will form a wireless adhoc network, where each device will attempt to forward any data packets to their destination over multiple hops. The resulting wireless sensor networks (WSN) such as the one shown in figure 1 have very different requirements to typical internet infrastructure, and so are currently a very popular area of research.

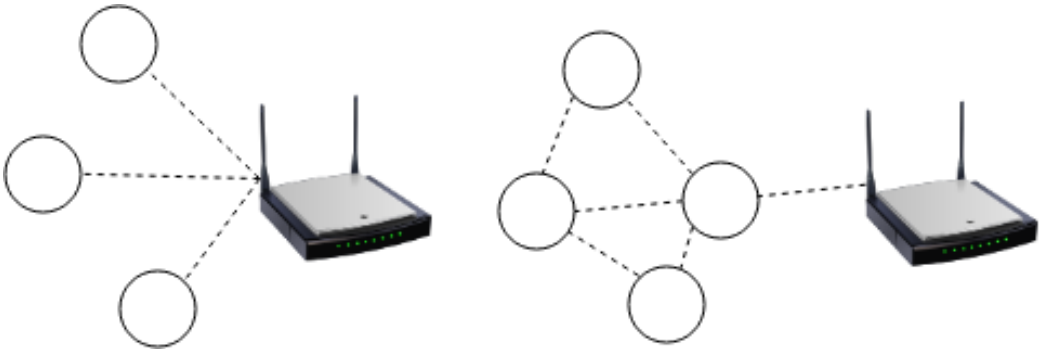

Figure 1: On the left are IoT devices connecting directly with a dedicated router, and on the right the IoT devices form a WSN in order to reach the router.

IoT devices are often restricted by battery life, memory, and computational capabilities. This makes reducing communication overhead and the amount of bookkeeping the main goals for most routing protocols in WSNs. However, most IP routing protocols focus on finding the shortest route between a source and destination, which often results in a small number of paths being heavily used and so some nodes are especially drained due to processing and forwarding of packets. WSNs also experience more mobility

than wired networks, which requires more update messages to be flooded throughout in order for packets to be routed correctly. This can result in a network partion once crucial nodes fail (due to loss of battery), rendering a section of still operational nodes useless. An approach to routing and addressing that reduced the networking overhead and attempted to balance traffic across several paths would allow IoT networks to remain operational for longer.

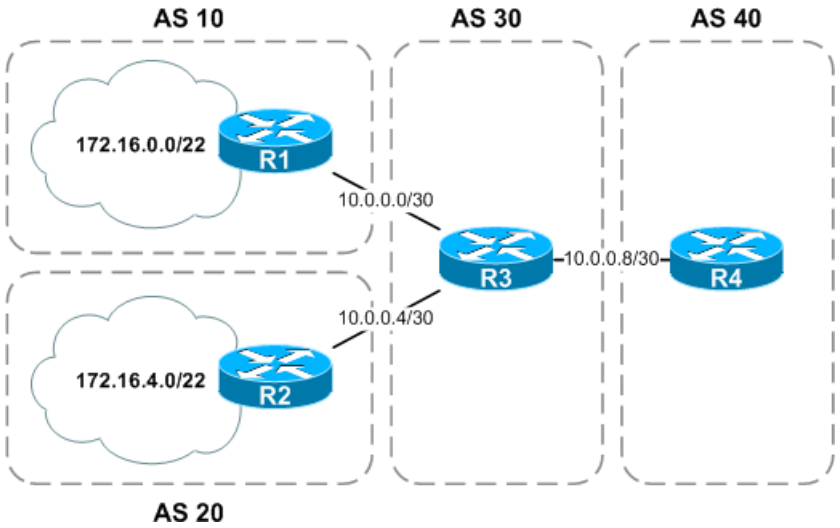

Figure 2: Example of address aggregation (Image from [1])

IPv4 is currently still the most used addressing scheme, and despite the imminent exhaustion of IPv4 addresses [2], IPv6 is being adopted slowly [3]. One of the main issues with IP addresses is the fact that they are used to both identify a system and to determine its topological location. [4] lists several of the downsides to this overloading of IP addresses, and why the protocol was still adopted despite them. For the variety of topologies and dynamic nature of WSNs, this overloading causes many issues. Usually when using IP, routers are able to perform address aggregation to reduce how many addresses they have to record for forwarding purposes, like in figure 2. But in cellular networks like those used by mobile phones for example, the path to the devices that packets need to take is constantly changing, and so mechanisms such as network address translation (NAT) and complex protocols such as Mobile IP need to be implemented.

Due to the inability to aggregate addresses, the scalability of the inter-

net is being challenged. An IAB workshop [5] detailed how the DFZ RIB databases are growing in size exponentially due to the increasing number of devices and an inability to perform address aggregation. One of the main causes of this inability is traffic engineering. In order to improve reliability, many ISPs are using multipath routing with the intention of balancing load. This improves network performance for the operators that use them, but with IP it places greater stress on the default-free zone (DFZ) routing information base (RIB). Multihoming is also being used to improve reliability, but with IP this requires routing entries to store multiple addresses for one host.

Given the difficulty involved in migrating from IPv4 to IPv6, it is very doubtful that introducing an entirely different protocol for the internet would be successful. A backwards compatible solution would likely be the only solution that would be adopted within a reasonable time frame. Since so many issues with IP are due to the overloading of the IP address, the alternatives that are being considered use an locator-identifier split approach.

Figure 3:

Both multihoming and mobility are far simpler to implement and maintain if the identity and topological locator of a host are separated, and this is how the Identifier-Locator Network Protocol functions. [6] proposes ILNPv6, which implement ILNP with the same address space as IPv6 and the same packet structure as IPv6, but with different semantics for interpreting the addresses. ILNPv6 splits the original 128-bits used for an IPv6 address into two 64-bit fields: the upper bits representing the locator and the lower 64 bits representing the identifier (figure 3). The version field in the IP header is used to differentiate between ILNPv6 and IPv6 packets, and routers that don't support ILNP can interpret the packets as IP without any issue.

The locator value identifies the subnetwork that a host belongs to, and a host can have multiple locator values, providing multihoming as shown in figure 4. The ID part of the address is unique to the host, and provides a fixed address which can be used at the transport layer. Because of this, ILNP is well suited to WSNs. Sensors are able to change locators without any changes

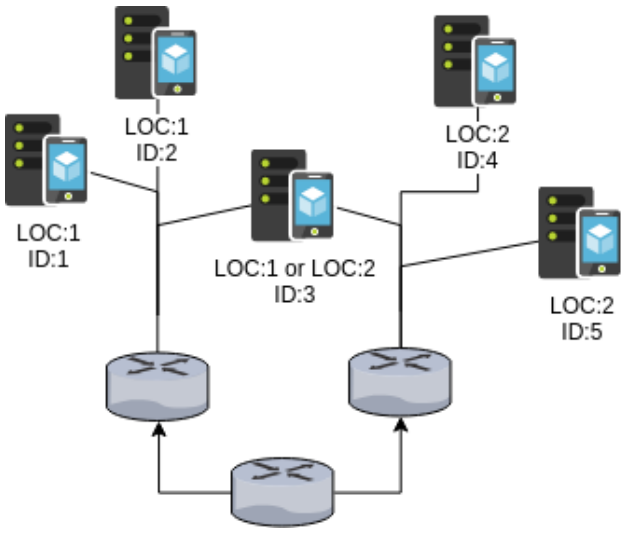

Figure 4: Example usage of ILNP addresses. The host with ID 3 is multi-homed but maintains its ID in both locators whereas IP would require two different IP addresses.

to their transport layer address, and routing tables can store paths to locators rather than attempting to aggregate addresses in a dynamic topology.

ILNP requires some additional information to be made available via the domain name servers (DNS) in order to function at scale. Since an ID is provided at the transport layer to identify a host, DNS need to provide a locator for routing the packet to that ID. In figure 5, the steps taken to resolve an ID to a locator are shown. Though an ID is used in the example, ILNP encourages the use of fully qualified domain names (FQDNs) instead to complete the network layer abstraction. An FQDN can be resolved to an identifier and locator also.

1. Send to DNS request for the locator of the host with ID = 5
2. Router forwards request to DNS (or replies from cache)
3. DNS responds with the locator for ID 5. If the end host is multihomed it can reply with multiple locators and a preference value.
4. Router forwards reply to requesting host.

5. Host is now able to construct the ILNP packet and send to the destination.

6. Router uses the locator to determine where to forward the packet.

7. Packet arrives at destination.

In a WSN, the extra round trip to the DNS could be wasteful if just resolving IDs that are located within the WSN, and so a routing protocol would need to include a way to discover locators without the DNS.

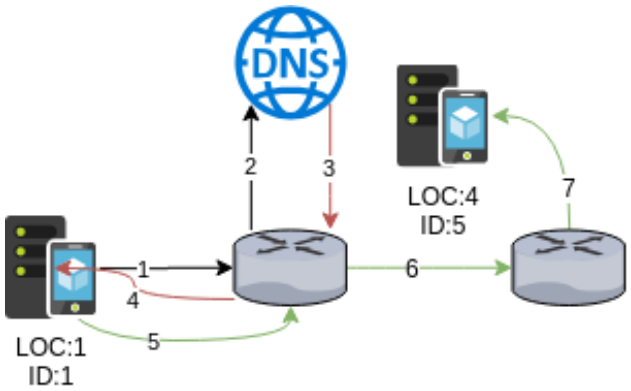

Figure 5: The steps taken by host with ID 1 to find the locator for the host with ID 5.

Though ILNPv6 is very backwards-compatible, there are still some difficult challenges involved in its deployment. [7] describes how the tight coupling of the C socket API and IP addresses could cause issues in some legacy applications.

The main goal of this project is to produce a library for a prototype communication protocol for wireless Internet of Things (IoT) devices based on the identifier-locator approach. By providing a libary for future developers and researchers to use, the adoptation of ILNP could be accelerated. With the ability to emulate networks that use ILNP, routing protocols could be designed that optimise for the likely imminent change in addressing scheme.

For the example routing protocol, an agricultural sensor network scenario was chosen.

A sucessful implementation would include:

1. **Load Balancing**: The protocol will attempt to evenly disperse load across a network of IoT devices. This could be measured by demonstrating a reasonably even distribution of packets throughout the network of nodes when simulating sensor traffic.

2. **Soft Handovers**: The protocol will attempt to handle 'soft-handovers', allowing nodes to move between subnetworks and handle node failure without a noticeable effect on performance. A successful implementation will recover from failed nodes, with messages still arriving reliably. This could be tested by emulating network traffic, and triggering nodes in certain positions in the network topology to fail.

3. **IoT Optimization**: The protocol will account for battery usage on devices to reduce energy drain on devices. This will be tested through emulation of an IoT network and the energy cost of packets.

4. **ILNP Addressing**: The protocol will be built around the identifer-locator split addressing scheme ILNPv6, and will perform better than IP in the same scenario.

2 Context Survey

Despite the intitial motivation for WSNs being military applications, they are now being used to solve many other problems.

The sensitivity of crops to changes in climate and agricultures crucial role on national economies has naturally resulted in large amounts of research and development. [8] describe how sensor technology is being used to monitor conditions in greenhouses, fields, and bodies of water. In order to make accessing this data more convenient and to help automate processes, they implemented a wireless network of enviromental sensors. This data would then be collected at a sink node and could be analysed from an application.

WSNs currently often require specialised applications and manually specified network configurations in order for consumers to collect and analyse the data they produce. In order to make access to the data more standardised and easily available, research is also being done to connect sensor networks to the cloud. [9] propose enviromental sensors for urban environments with gateways to to the internet which could integrate into our own homes. By providing cheap montoring and actuating sensors to the general public, they hope to encourage healthier eating habits, as people would be able to grow their own vegetables effectively.

[10] list the other applications of WSNs, and the different approaches to integrating WSNs and the internet. They recognise that providing a single gateway results in a single point of failure, and so focus on methods involving multiple or integrated gateways. This requires that sensors adopt the responsiblity of managing their networks topology instead of just forwarding towards a sink node.

Previous research involving integrating WSNs into the current internet infrastructure is typically structured around IP. For all the reasons mentioned earlier, researches have realised the benefits of building WSNs using other approaches such as the locator-identifier split.

In RFC6115 [11], ILNP was listed alongside several other solutions that resolved the issues faced by IP. One of the most well researched solutions listed is the Locator-Identifer Split Protocol (LISP), which has already been deployed in 60 sites over 10 countries [12]. Whilst it does not have natural support for network mobility, attempts have been made to provide it that require further extensions to the protocol [13].

ILNPv6 on the other hand has been able to demonstrate reliable mobility

using a soft handover process [14]. Soft handover is where a host remains connected to it's original network while transitioning to a new one, which avoids loss of data when packets are still being routed to it's original network.

Soft handovers are crucial for high mobility devices such as smartphones, which are constantly transitioning between networks. Handovers are implemented in Mobile IP for IPv6, and has been improved since IPv4, but the method is still not as performant as it could be. It also further muddies the meaning of IP addresses, due to the use of different addresses (e.g. 'home' and 'care-of' addresses) in order to redirect packets to the mobile node.

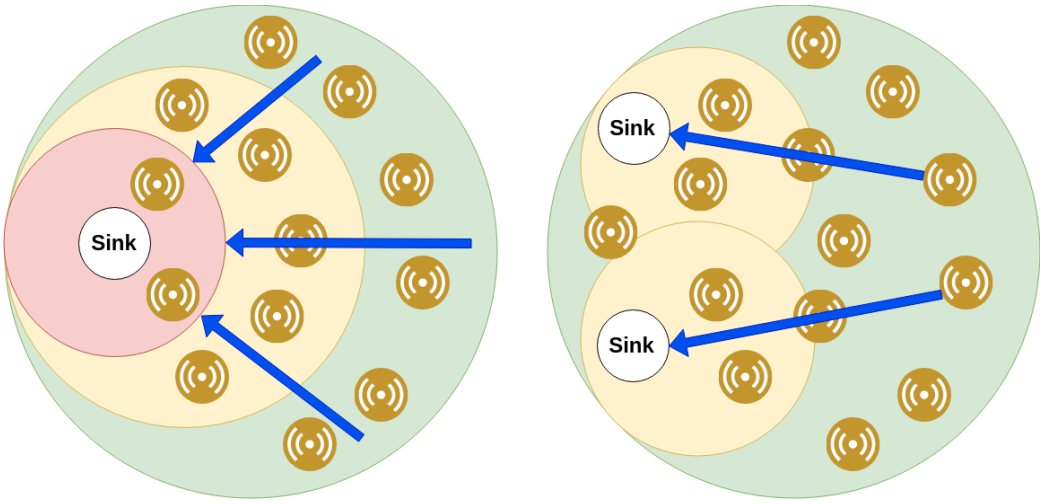

Figure 6: Comparison between the use of a single sink and multiple sink nodes. Coloured circles represent the amount of traffic (red, yellow, green where red means the high traffic) and the small yellow circles represent transmitting devices.

Multihoming is also incredibly useful for WSNs, as it can potentially allow them to scale without being reconfigured. WSN lifetimes are inversely proportional to their diameter when a single sink or gateway is available [15]. By providing multiple sinks, the levels of traffic around the sink nodes is decreased, increasing the longevity of the network. With ILNP the end destination identifier could remain the same, and so the underlying routing protocol would only have to include a way of realising a route to the new interface like in the scenario shown in figure 6.

A different addressing scheme is only part of the solution required for

successful intergration of WSNs and the internet. Due to the myriad of situations that IoT devices are used, there is no one-size-fits-all solution, and so many versions of IoT routing have been proposed.

[16] classifies different routing protocols and provides a survey of protocols for each class. Routing protocols are either proactive, reactive, or hybrid.

Proactive protocols maintain knowledge of the network topology using regular flooding and beacon messages. This is acceptable for most of the internet infrastructure, but periodic messages can drain precious energy from devices that are powered by battery. Since the state of links are always tested, packet delivery is more reliable, and if nodes change position a new path can be built quickly. In the current internet, OSPF is one of the most popular interior gateway protocols, and so naturally researchers tried to adapt it to wireless networks. This produced the Optimized Link State Routing (OLSR) protocol [17] and Open Shortest Path First MANET Designated Routers (OSPF-MDR) protocol. However, in low mobility scenarios reactive protocols perform far better than either of these options.

Reactive protocols involve only seeking out routes to destinations when one is required. Examples include AODV [18] and DSR [19]. These protocols were designed for mobile ad-hoc networks, and so most research involves improving them to be aware of energy availability. For example, [20] produced a solution that considered the mean energy of a path before choosing which route to suggest to the requesting node. This would discourage the use of paths where nodes had low energy reserves. By doing so, less route repairs need to occur, and network partions are postponed. Reactive protocols are also well suited to the wireless medium, as the lack of periodic updates reduces the chance of interference which exists with proactive protocols.

In order to balance reliability with overhead, a hybrid approach can be applied. For example, [21] use a clustering approach in order to manage resources within subsets of nodes. Since not all WSNs are homogenous, this approach can take advantage of some nodes having larger energy reserves than others by electing these nodes as the coordinator of the subnetwork. [22] also implement a hybrid routing protocol for vehicle mobile networks which uses beacons to monitor link states between nodes to allow recovery to occur when the routes created by the proactive protocol fail. They use GPS to help route packets when their target is in a known location. Both these implementations use IP, but grouping nodes by locator and calculating locators on GPS coordinates could be done with ILNP in a more effective way, which is why providing an emulation tool for ILNP ad-hoc networks

would be beneficial to research.

3 Protocol Design

3.1 Motivation and Overview

The protocol used is a reactive zone-based routing protocol (ZBR), and is based on the work by [23] with added support for energy awareness. It was designed with the following assumptions:

1. All nodes have equal computation and network ranges.
2. All nodes that share a locator are also physically located in a simliar area.
3. Locators cover a similiar physical area, in a lattice structure.

The scenario this protocol was designed for involved agricultural sensor networks. These often consist of many sensor nodes for monitoring the environment and a few actuators for triggering sprinklers or covering delicate crops. Based on the data collected by sensors, they could control the actuators within their locators themselves, whilst reporting their readings to the sink for further analysis. By splitting sensors into geographically relevant zones and assiging locators using tools such as GPS, the arriving data could also be grouped by origin location to help with visualisation and analysis at the sink.

Figure 7 provides an example of communication where a sensor wants to trigger some behaviour at an actuator node, and also sends its readings to the sink node.

In this scenario, the only knowledge that would be available upon initialization of a sensor is its ID and locator, and the ID of the sink node. Sensors would detect any other hosts within their subnetwork, which could be devices such as sprinkler systems, shelters, or cameras. If the sink node was not part of their subnetwork, then they would need to locate it.

In order to scale for large farms, this zone based protocol uses a different routing strategy for routing within and between locators, referred to as the intrazone and interzone routing protocols (IARP, and IERP). The IERP is based on the reactive protocol AODV, and the IARP is based on the link state protocol OLSR with added fields to try and account for energy usage in paths.

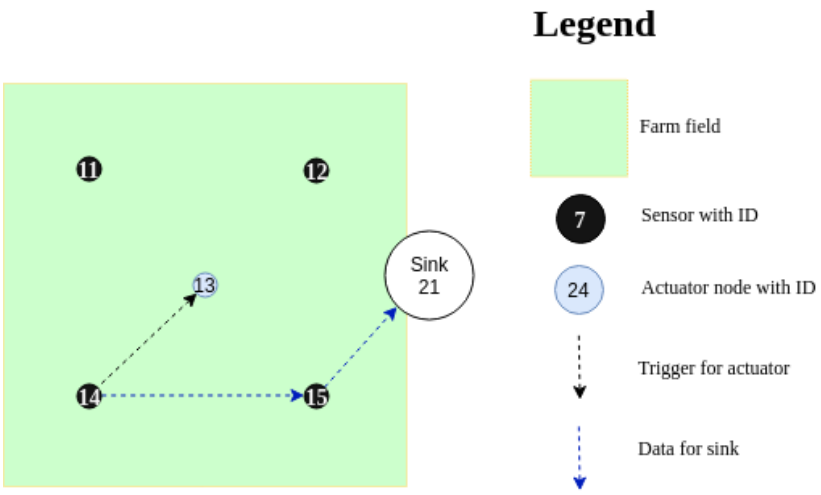

Figure 7: Communication patterns in agricultural WSN

A reactive protocol was chosen for communication between locators in order to limit the amount of network knowledge required for routing. It also has the benefit that link breakages do not need to be broadcast to the entire network, and can instead only be sent to nodes relying on that links existence at the time of breakage.

A proactive protocol was chosen for routing within locators as it would provide node discovery, and would hopefully reduce the number of repeated route discoveries after breakages by being able to find alternative routes within locators. Whilst the keepalives may affect battery life, the static nature of such a network would allow for a large delay between them, making their effect neglible.

3.2 Intra-Zone Routing Protocol

Initially, each node broadcasts its presence to its neighbours in order to learn what links were available to it using a *Hello*.

Neighbouring nodes reply with their current link state database (LSDB) in a *LSDBMessage*. This message contains the neighbouring nodes knowledge of the internal (within the locator) topology, as well as the links to external networks (other locators). Whilst still in the intiliziation phases, neighbours will flood their entire LSBs to their neighbours until no more changes are

detected. At this point they can be considered initialized. This ensures that every node in the network has the same database and is aware of all other nodes in the network.

The LSDB messages also provide a metric for choosing paths based on their energy levels. Each node calculates a value λ using equation 1 and include it in initial *Hello* message, which is then included in each link description in the LSDB message so that it is propagated throughout the locator. The equation for λ is visualised in Figure 8.

$$\lambda = 1 - (1 - E)^2 \tag{1}$$

where E is the % battery remaining. It is based on the equation for λ in [24], adjusted so that $E = 0$ occurs when the battery is empty. The effect of decreasing E was chosen so that the value of λ would be similiar during network initialization, and would gradually become important as the E decreased. If the value was to change quickly, it would increase the possibility of route flapping, which is where a router will repeatedly forward packets along two seperate paths alternately.

[24] also included a load balancing factor that would account for the number of neighbours a node had, and this was considered but was not implemented.

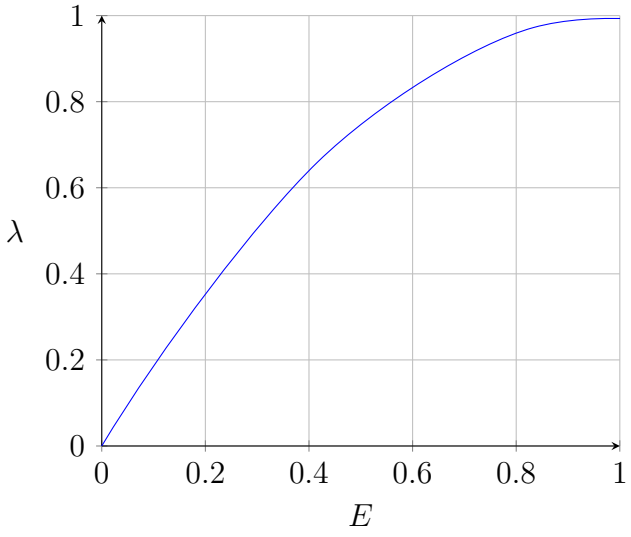

Figure 8: Plot of Equation 1

Nodes that exist on the border of locators will discard LSBs they receive from the locators they are not part of. They will however use these messages to learn what other locators can be reached from their neighbours, eventually providing them with next hops for all locators adjacent to theirs in the WSN.

Once converged and the entire local topology is available, data packets would be routed using the following logic:

1. If the packet destination ID matches the current node, then it will add the payload of the packet to the received queue.
2. Otherwise it will attempt to find the next hop for the packet, provided by the forwarding table, and forward it to the node with that ID.
3. If no entries are found in the forwarding table:
   (a) If the packet destination locator matches this nodes locator, then not finding a next hop suggests that the destination ID doesn't exist, and so the packet is discarded.
   (b) Otherwise, the inter-zone routing protocol will be invoked, but only if the packet originates from the current node.

This method ensures that packets arriving from external locators will receive best effort routing based on the assumptions that:

1. All nodes within a locator have at least one path between each other.
2. Data packets are only sent to other locators once a route has been established.

With the link state databases synchronised, nodes then begin periodically sending keepalive messages. Figure 9 shows how these messages are processed.

Whenever a link is detected as being lost due to the lack of keepalive messages after a period of time, then this information will be shared within the network. A *ExpiredLink* message will be broadcast, and flooded by nodes within the same locator in order to remove the link from their network, or to inform the other nodes that a neighbouring locator is no longer accessible via a given link.

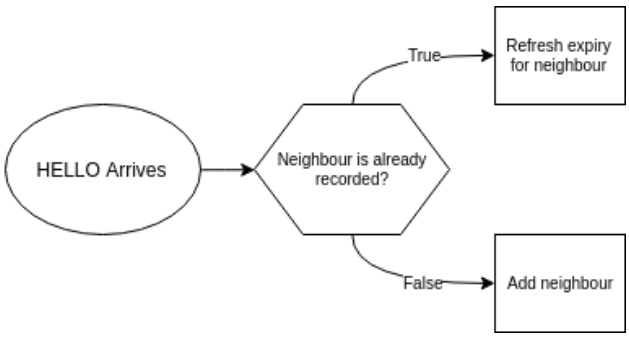

Figure 9: Keepalive (hello) flow.

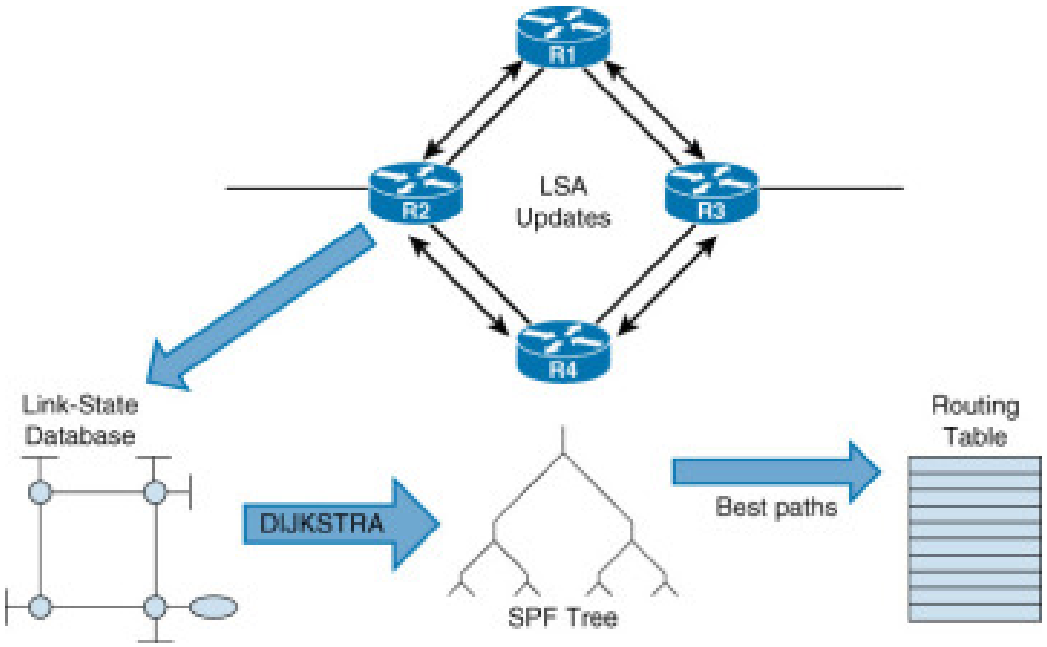

Figure 10: Intra-zone routing protocol process (image credit [25])

The internal forwarding table is constructed from the link state database messages as shown in figure 10. The network graph is converted to a forwarding table by constructing a shortest path tree. In the case where two equal length paths exist between this node and a destination, the one with the best value for λ will be chosen for the next hop. The forwarding table also caches the locators for nodes outside of the current locator so that route discovery is not repeated unnecessarily.

3.3 Inter-Zone Routing Protocol

The inter-zone routing protocol is based on AODV, which is a reactive protocol. AODV typically has three phases (route discovery, maintenance, and repair), but in this implementation route discovery also provides locator discovery.

Locator discovery is necessary since locators are not known for all nodes in the network. Usually the DNS would provide the locators for a given identifier or fully qualified domain name (FQDN), but since that is not a practical option for a wireless sensor network, the locator will have to be found manually.

To discover routes to a given ID, a route request is made. A node will create a route request and forward a copy towards each of neighbouring locators. The first node that receives the request in each locator will check if the destination ID is in their subnetwork, and if so they will reply. Otherwise, they will forward it to any neighbours that have not already been added to the hop list. The route discovery process is summarised in figure 11, where a node from the leftmost locator (1) is requesting a route to a node in the the rightmost locator (4).

Figure 12 shows how RREQs are processed at each node. The destination node replies to all route requests for it, as this can provide multiple paths to the requesting node. Intermediate nodes however only forward requests based on whether or not they've seen them already. This can be established based on the request ID in the RREQ message, which coupled with the source ID in the ILNP packet header can identify duplicate requests. Usually in AODV, route requests are flooded to all nodes. In this implementation, route requests are unicasted to border nodes instead in order to reach other locators.

When using ILNP instead of IP with AODV, packets can be routed based on the identifier alone, and this can result in multiple paths to the same node. This provides robust communication if the end destination is multihomed (i.e. has interfaces to multiple locators), with less complexity than in IP.

Once the RREQ reaches its destination, or any intermidiate node that already has a path cached to the that destination, a Route Reply (RRPLY) is generated by copying the full path from the RREQ or route cache and sending it back along the same path it arrived to the requesting node. Figure 13 shows the processing that occurs at each node.

By forwarding RRPLYs along the reverse of the path that they contain, we ensure that the route hasn't broken between creation and reception of the

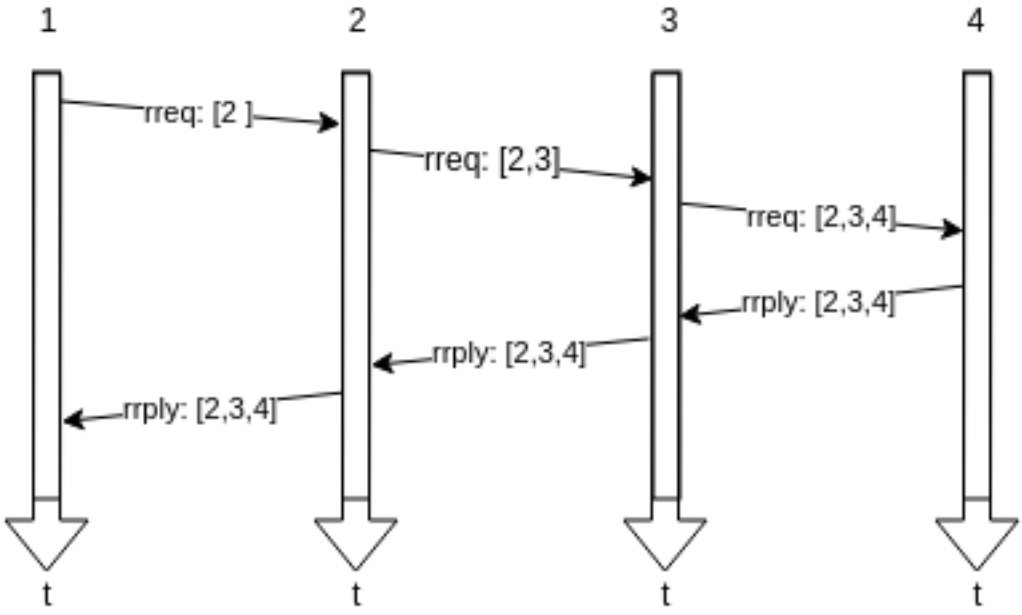

Figure 11: Route discovery made by a node in locator one for a node that is found in locator 4.

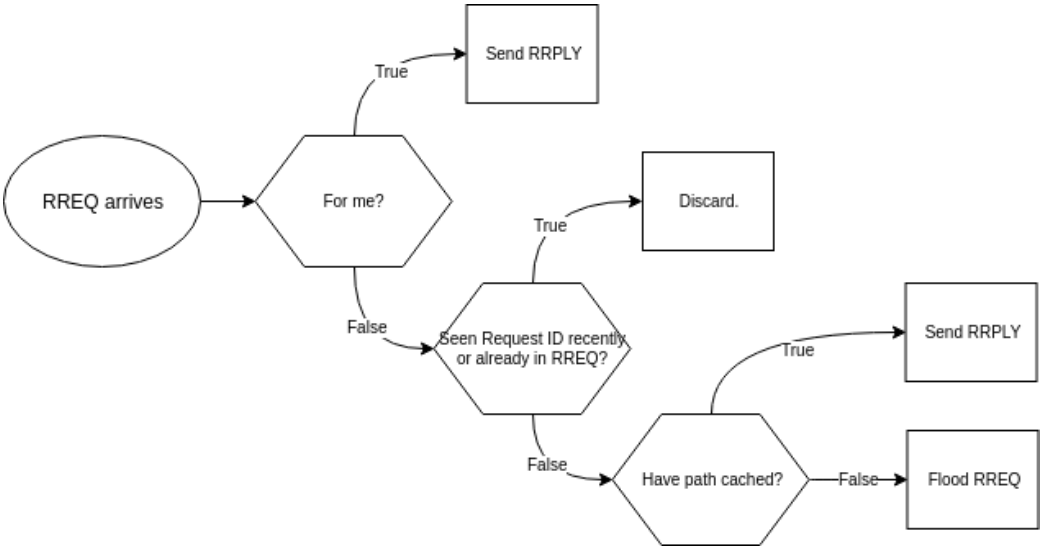

Figure 12: Route Request (RREQ) Flow

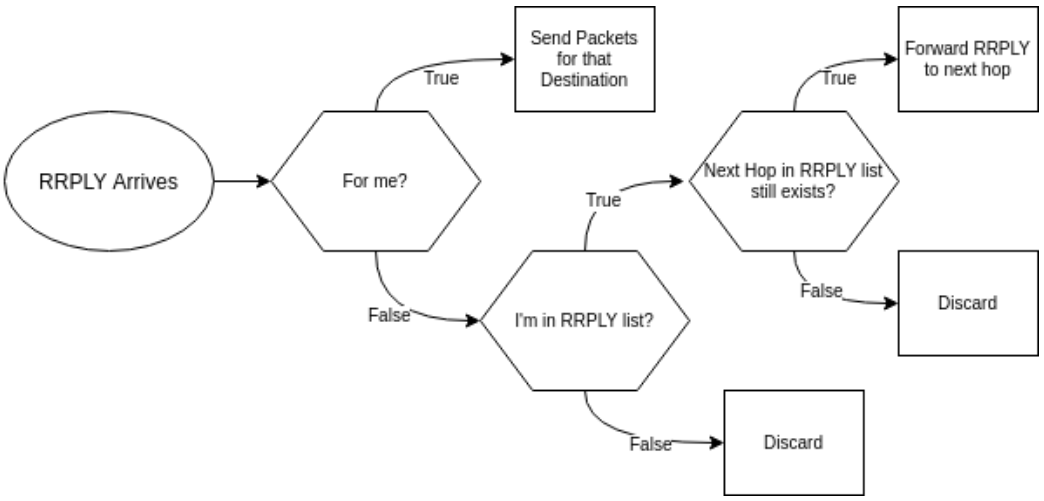

Figure 13: Route Reply (RRPLY) Flow

route request. Intermediate nodes can filter erraneous route replies by only considering those where they know the next hop neighbour is still available.

This only works if we assume that all links are bidirectional which is not always the case especially in heterogenous WSNs, due to differences in transceiver ranges. [26] shows that accounting for unidirectional links in a protocol does not provide much benefit compared to the increased overhead. Also in our scenario, the nodes are likely to be homogenous and evenly spaced, so transceiver power can be assumed to be equal throughout, with no interference hot spots.

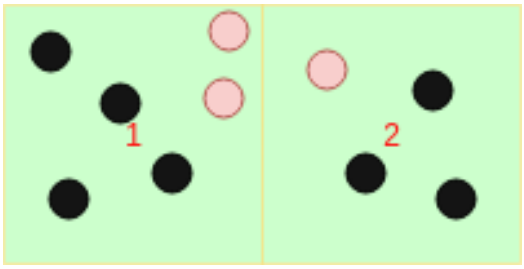

Figure 14: Border Nodes (Red)

If multiple border nodes (nodes that have links to another locator) provide links to the same neighbour zone, then they will will alterate between them

using a weighted round robin method based on the path costs to avoid all traffic travelling through the same border node. For example, in figure 14 there are two links from locator 2 to locator 1, and so either can be used.

When packets arrive to be forwarded to another locator, the interzone forwarding table will be used to find the next hop locator. Then the intrazone forwarding table will be used to provide the next hop ID in order to reach a border node that can get the packet closer to the destination locator;

Figure 15 shows the full locator discovery occuring, which returns the locator of the destination node with the requested ID, and the hop by hop list of locators that have to be crossed to reach it. Once this locator discovery has been carried out, border nodes will be aware of how to reach each locator, and so will be able to forward packets to that destination.

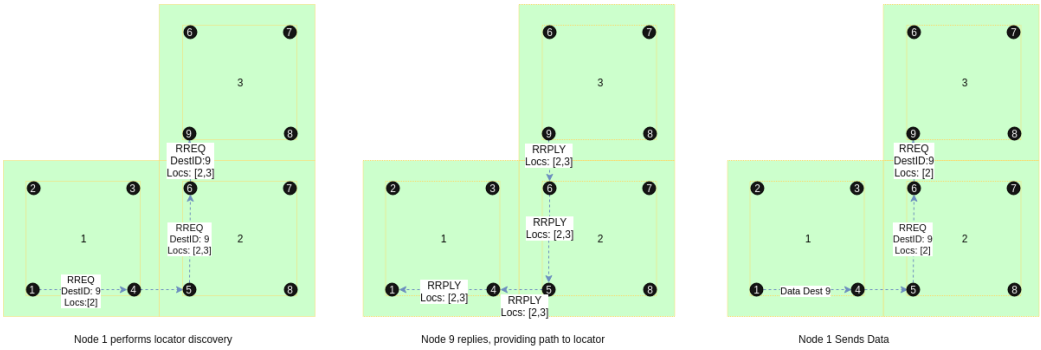

Figure 15: Locator Discovery where node 1 needs a route to node 9. The black numbered circles each represent a node with an ID, and the green squares are locators.

If the keepalive process fails for a neighbour and two locators in the route are no longer linked, then a node will forward a RERR to all previous nodes in any paths it takes part in, shown in figure 16.

Any packets that were sent between the failure of the link and the receiving of a route error will most likely be dropped, and future packets will be delayed as route discovery will have to take place again unless this node is aware of a disjoint path to the destination.

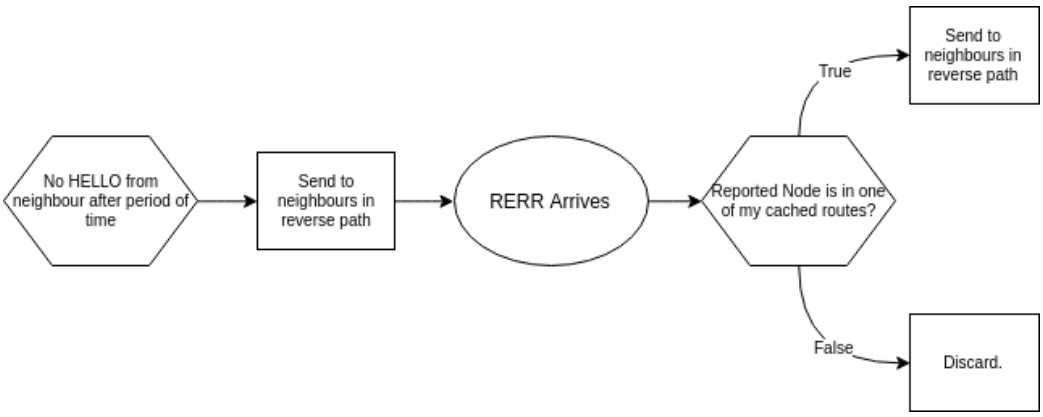

Figure 16: Route Error (RERR) Flow

4 ILNP Testbed Implementation

Due to the recency of ILNPv6, there does not currently exist software for emulating wireless ILNP networks as there is for IP, therefore a custom testbed had to be implemented.

The testing environment was implemented using Python 3.7. The project has three main components: The wireless network emulation, routing and network management, and sensor emulation. Figure 17 shows how each of these components correspond to a section in the network model.

4.1 Wireless Network Emulation

Emulating wireless communication was achieved by building an overlay network using UDP with multicast. A single UDP socket provided a communication endpoint.

A multicast group exists for each node ID, and so neighbours of a node would join the multicast groups of their neighbours in order to receive broadcasted packets. Figure 18 shows the ranges of each sensors radio signal as dashed lines, where each node within the range joins the multicast group.

IPv4 uses the Address Resolution Protocol (ARP) and IPv6 uses Neighbour Discovery (ND) in order to produce a mapping between IP addresses and the link layer addresses of other machines in a local network. In our overlay network, the IP addresses of the underlying network could be treated as MAC addresses. This would allow us to emulate neighbour discovery in

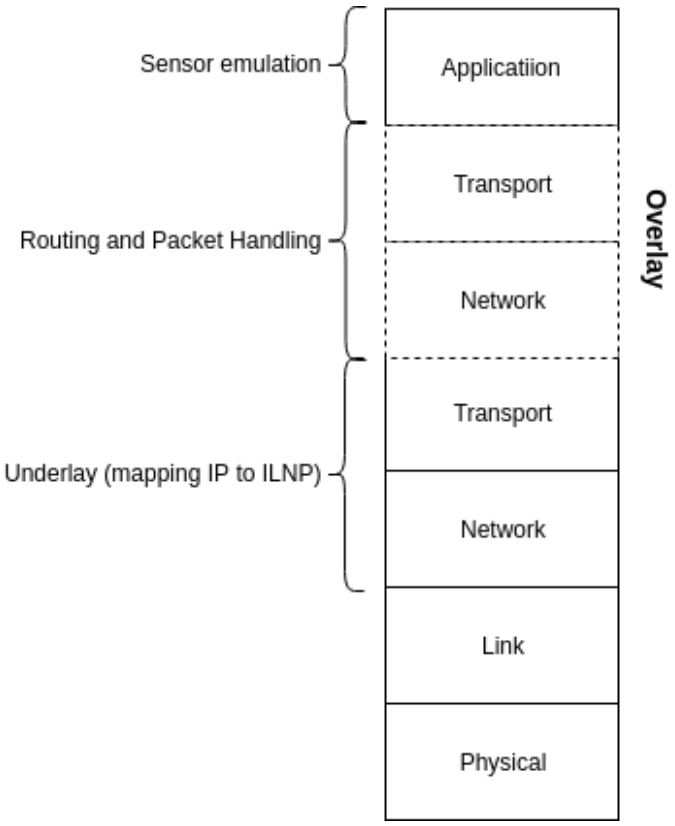

Figure 17: Structure of project with regards to the network model.

ILNPv6, and provide unicast communication once a mapping between these virtual link layer addresses and ILNP IDs were obtained via broadcasted messages.

The network interface abstraction provided the following methods:

1. **send**: For unicast communication.
2. **broadcast**: For establishing neighbours or flooding packets.
3. **receive**: For receiving packets, broadcast or unicast.

A daemon thread continuously polls the network interface for incoming data. It then parses the contents of the packet and records the source IDs and source IP addresses of any neighbourhood discovery packets for later communication. Each message type has its own class with its own implementation of

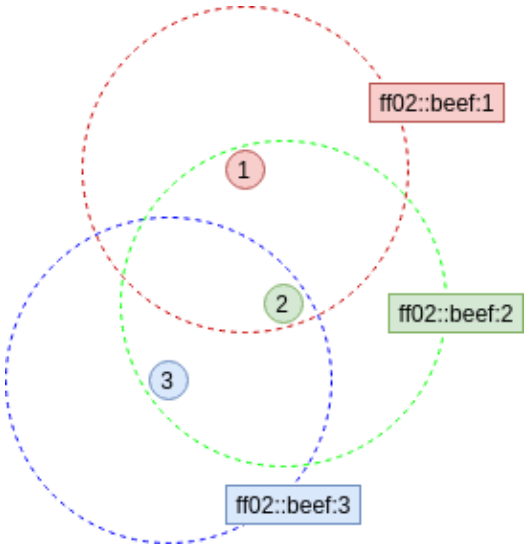

Figure 18: Wireless broadcast emulation using multicast

__bytes__() and *from_bytes()* which would allow them to be serialized from the arriving bytes. To reduce copying and speed up array splicing, *memoryviews* were used often.

All polling functions included a timeout in order for each thread to check if it has been asked to terminate, if for example the power levels had reached a critical level. If so, a monitor object shared between all threads would be used to trigger a clean shutdown.

4.2 Routing and Network Management

The router polls the queue of parsed packets and decides the next course of action. It attempts to route data packets using the current forwarding tables, and hands off control packets to the control plane.

The control plane handles any control packets or any packets that can't be forwarded using the current forwarding table, and is responsible for populating the forwarding table. It also has a daemon thread which performs maintenance tasks such as aging and retrying route requests, sending keepalives, and monitoring links with neighbours.

The interzone module provides the handler methods for the inter-zone routing protocol. It buffers packets waiting for a route reply for a given des-

tination and handles the forwarding and processing of any inter-zone protocol messages.

The ILNPPacket class when serialized is identical to that from the ILNPv6 RFC. In order to differentiate between control packets and data packets, the payload of the ILNPPacket would be wrapped in a ControlMessage class. Typically control packets and data packets would arrive on seperate ports, but this was not the solution adopted for this implementation.

4.3 Sensor Emulation

The sensor initialises the ILNPSocket, and depending on whether or not it is configured to operate as the sink (where all the data from every node is collected) begins either polling for packets or sending 'readings' at intervals.

To mock realistic application data, a random fluctuation is applied to a series of values that would be relevant in our scenario: temperature, humidity, pressure, and luminosity [8].

The monitor object shared throughout the project tracks any packets that are sent, and the sink also tracks any readings that it receives. These are then written to a file once the socket is closed for analysis.

5 Experiment

The aims of the experiment were to show that protocol succeeded in meeting the goals described in the introduction. This was be achieved by simulating a wireless sensor network for monitoring fields in a farm. Figure 19 shows the layout of the scenario being tested.

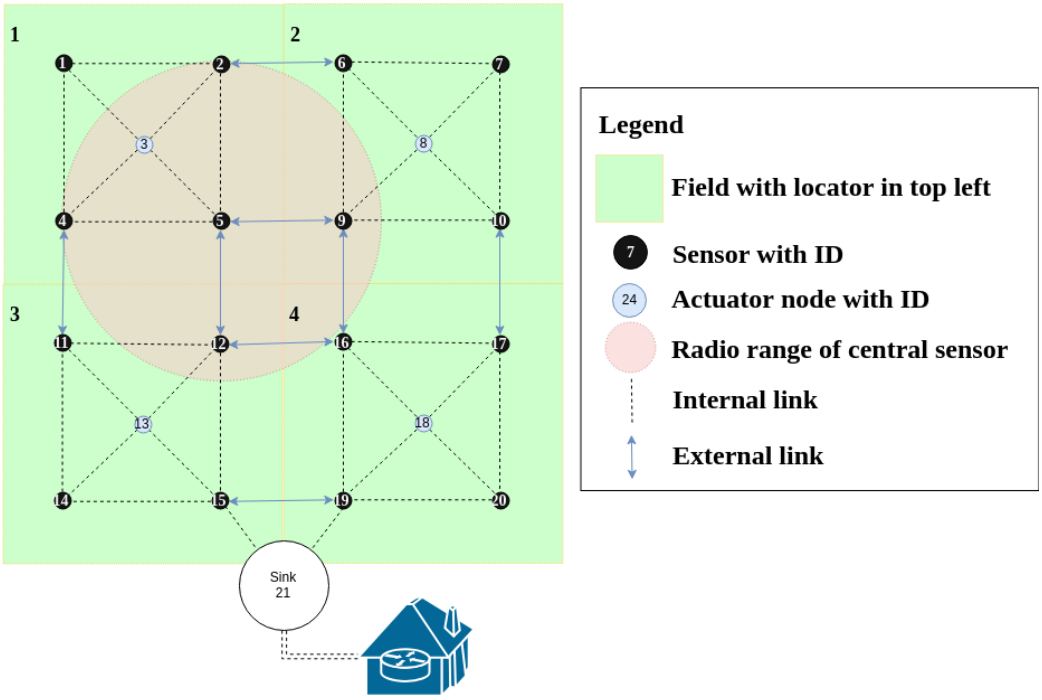

Figure 19: WSN layout with example routes.

For the emulation, we assumed that:

1. Each sensor had equal radio ranges.
2. Sensors always listened for packets with same antennae (i.e. no lower power antennaes would be used until a signal was detected[27])
3. Sending a packet has a fixed energy cost, regardless of size of payload.
4. Collisions and interference would not occur.

Two experiments would be run in order to compare performance when using ILNP and IP. Figure 20 shows that ILNP can emulate IP by either:

1. providing a locator for every node, essentially treating every node as a subnetwork.

2. using the same locator for all nodes, which would require knowledge of the entire subnetwork in order to route correctly.

Both extremes of the spectrum will provide identical flat addressing schemes like IP, but would behave differently under the implemented protocol. Using a locator per node basically reduces to AODV, and using a single locator would reduce to OLSR. Figure 20 shows the effect of changing the ratio of locators to nodes.

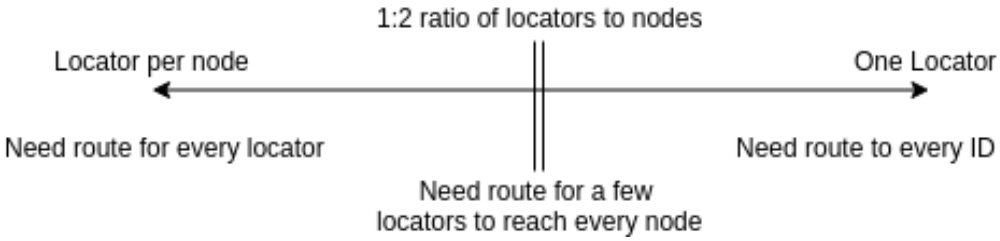

Figure 20: Behaviour as the ratio of locators to nodes is changed.

5.1 Setup

The experiment was run on the lab machines, remaining within the schools network.

To construct and automate the testing environment, a series of bash scripts were written. The first of which would perform a quick liveness check by pinging a list of lab machine IP addresses and constructing a list of the IP addresses that responded.

Since multicast was being used to emulate wireless broadcast, my user ID was used as the second-last octet to avoid collision with other experiments possibly running on the network. This is sourced by the implementation and does not need to be specified in the configuration.

The configuration files that were used to initialize each sensor node were generated by performing string replacement on a template configuration.

"LOC_1_ID_2_MCAST_1_2" for example would produce a configuration file for a node with ID 2, locator 1, that was part of the multicast groups "ff02::dead:1" and "ff02::dead:2". Being able to describe network topologies using single strings allowed multiple tests to be configured easily, and possible future work could involve generating these configurations using a visual representation of the network.

With the list of available computers and the configuration files generated, another script was written that would *ssh* into the lab machines one by one and start the sensor emulation program with each configuration. Logs from each instance of the program were redirected to a file based on the node ID for monitoring the network.

5.2 Method

In order to monitor the type and number of packets sent and forwarded by each node, an additional *monitor* module was added to the testbed implementation. For each packet sent, the monitor recorded a the ID of the sensor, timestamp, the type of packet (control or data), and whether or not the packet originated from this node or was being forwarded. Using this information, the flow of packets over time could be visualised, and the ratio of control packets to data packets (i.e. overhead) could be measured.

The sink node would also record each data packet it received in order to measure loss. Loss would be calculated from the difference between the number of data packets sent by each sensor and the number received by the sink.

Each test was run multiple times in order to ensure that the results were consistent.

5.3 Results

ILNP

Figures 21 to 26 show the concentration of data and control packets throughout the network over the period of three equally sized snapshots.

In figure 21, node 19 is able to immediately send packets to the sink and so it shows the highest data packet traffic. From the log files it is also evident that it is forwarding packets from the adjacent nodes in its locator. Figure 22 shows an increase in control packets from nodes in locators further from the sink, as they need to perform locator discovery to find the sink node. In

the next snapshot, we can see the effect of route replies as large burst of data packets and route replies are being forwarded by nodes 5 and 9. In the final snapshot, we can see that node 19 has failed. The rest of the network does not appear to recover as node 15 (the only remaining link to the sink) is not forwarding any more traffic than before.

IP

Figures 27 to 32 show very different traffic patterns to the ILNP experiment. The control traffic in each snapshot is fairly consistent due to the keepalives, though in the first snapshot the distribution of LSDB messages is not even. Data packets are being sent by each node but mostly for three individual nodes. These abnormal peaks in traffic are likely due to implementation error.

Discussion and Evaluation

The results do not give much insight into differences between ILNP and IP. They also do not seem to suggest any load balancing or recovery is occuring successfully. However, from the logs it was observed that the sizes of the forwarding tables and network graphs stored by each node are very different. In ILNP, only one locator worth of adresses and a path to the sink is stored, whilst in the IP experiment, both the network graph and forwarding table are large. If the WSN was to be scaled up further, this issue would be exacerbated. The limited memory of IoT devices would place a limit on the size of the network, whilst in this implementation the forwarding table would only be affected by additional nodes within the locator.

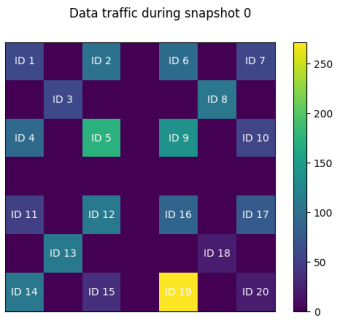

Figure 21: Data packets sent by each node during network initialisation when using ILNP.

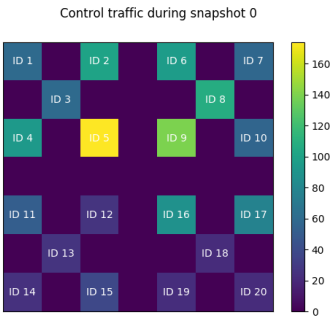

Figure 22: Control packets sent by each node during network initialisation when using ILNP

Figure 23: Data packets sent by each node once initial routes have been established when using ILNP

Figure 24: Control packets sent by each node once intial routes have been established when using ILNP

Figure 25: Data packets sent by each node once failures have started to occur when using ILNP

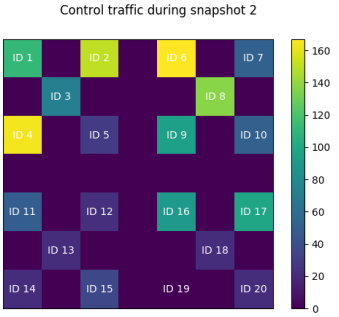

Figure 26: Control packets sent by each node once failures have started to occur when using ILNP

A different approach to the experiment had been considered before using only AODV, and the source code for this will be included in the submission. Whilst that implementation did perform better overall when routing packets in the same network topology than the implementation chosen, it did not make use of the locators in ILNP. The implementation used in this submission was used instead despite being less functional due to the fact that it provided a way to compare IP and ILNP (by changing the ratio of locators to nodes). Given more time to test and debug, the protocol could potentially perform very well, but the more interesting design was realised too close to the submission date which left little time for fixes.

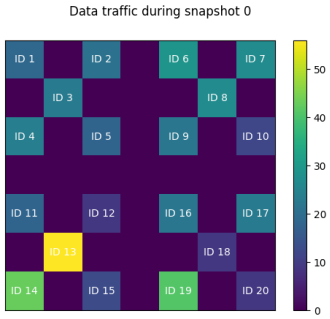

Figure 27: Data packets sent by each node during network initialisation when using IP

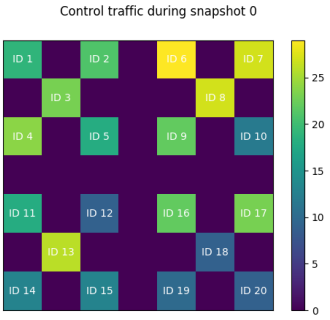

Figure 28: Control packets sent by each node during network initialisation when using IP

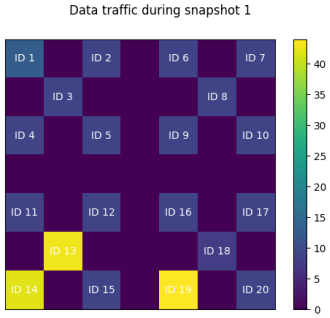

Figure 29: Data packets sent by each node once initial routes have been established when using IP

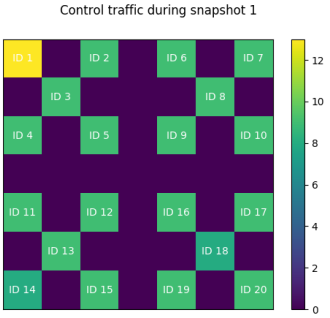

Figure 30: Control packets sent by each node once intial routes have been established when using IP

It could be argued that this experiment is more a comparison of protocols rather than of the two addressing schemes. WSN protocols have already been proposed that try to organise nodes into clusters and behave very similiarly to this protocol once these clusters are established. With IP however, these protocols require their own addressing schemes which are an abstraction of the underlying IP addresses. ILNP naturally supports grouping in protocols which means this abstraction would not be necessary. By reducing the complexity, protocols become easier to verify, test, and maintain.

Compared with a 'hot potato' routing algorithm, where packets are simply

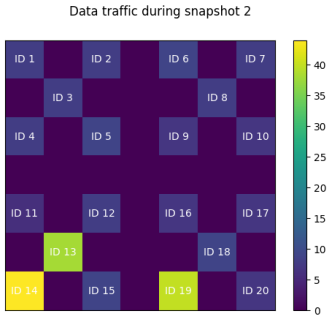

Figure 31: Data packets sent by each node once failures have started to occur when using IP

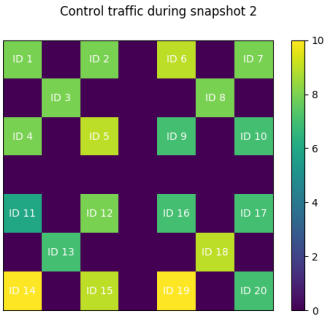

Figure 32: Control packets sent by each node once failures have started to occur when using IP

forwarded to a random next hop other than the one they arrived on, the algorithm should perform better since packets are at least directed towards the sink without loops. Whilst this is a low bar to set for a routing algorithm, it is a useful heuristic for determining if the complexity of a zoned routing protocol is actually beneficial.

A criticism that can be made of the chosen protocol is that its complexity made development and debugging very difficult. In order to test that it worked correctly, many scenarios had to be recreated. Since arrival times of messages would vary, finding the series of events that led to a problematic state was very time consuming. Verifying the protocol by applying exhaustive model checking would have definitely helped.

The protocol also may not perform well when locators do not have similiar structures. Figure 33 provides an example of how not being aware of the topology within neighbouring locators could result in very ineffecient routing paths.

Security was not considered at all when desigining this protocol, which definitely makes it unsuitable for many situations. It relies heavily on all control messages being legitimate and on the honesty of nodes regarding their ability to forward traffic. For example, if a node with plentiful energy wished to process less network traffic, it could advertise itself as having low energy reserves and be the least likely candidate for forwarding. Any host could also reply to all route requests it sees with a path to a locator it wishes to drain of resources.

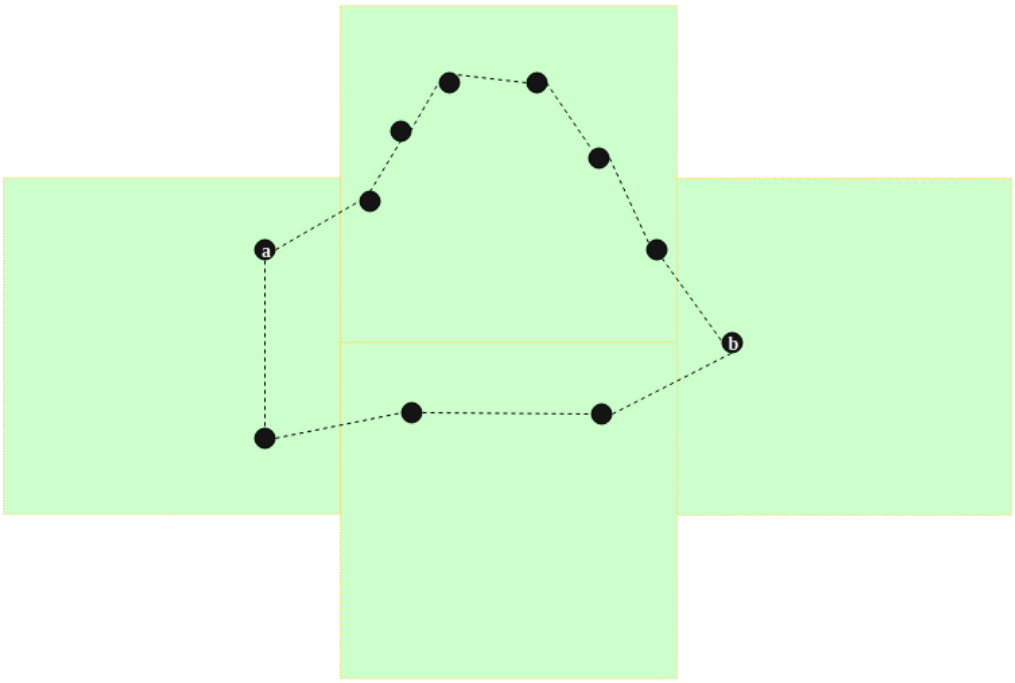

Figure 33: Node A wants a path to node B, and would likely consider the path through the upper locator since that link would bring it one locator closer to its destination, whilst the alternative requires another internal hop. Symbols have same meaning as in figure 19

6 Conclusion

Overall, the project was able to meet two of the original four goals. ILNP addressing was adopted successfully, and a protocol optimized for IoT devices was implemented. Due to issues with the end product, the experiments were not particularly useful for comparing ILNP and IP, but did allow for some discussion. ILNP required far less knowledge of the global networks state compared to IP, which could potentially allow WSNs using it to scale further than when using IP.

Future work that could improve the robustness of the protocol would be to handle the situations where a network partition occurs within a locator. One possible solution would be to allow locators to form dynamically. The implementation used originally featured a mechanism for 'joining' a locator instead of having them statically assigned as in the experiment. By identi-

fying the central node within a locator and setting a maximum radius for each locator, partions within locators could be resolved by both sides of the split attempting to merge with the surrounding locators. This was scrapped from the project due to a shortage of time, but attempts can be found in the version control history.

The project suffered due to taking too long to settle on a design, but the process of considering and attempting different variations on ad-hoc routing protocols was very educational, and has greatly strenghthened my understanding of networks. Whilst disappointed that I was not able to get a fully working prototype, the opportunity to work on and resesearch the potential future of the internet proved very interesting and enjoyable.

7 Appendix

Running the experiment

1. Update file paths in
   (a) *cleanup.sh*
   (b) *run_configs_on_machines.sh*
   (c) *run_with_config.sh*
2. Run *get_up_machines.sh* to ping all lab machines.
3. Run *generate_config_from_list.sh configs/_list.txt name* to generate python configuration file (.ini)
4. Run *run_configs_on_machines.sh configs/_list.txt name* to start application on list of machines.

Configuration for ILNP Test

LOC_1_ID_1_MCAST_2_3_4
LOC_1_ID_2_MCAST_1_3_5_6
LOC_1_ID_3_MCAST_1_2_4_5
LOC_1_ID_4_MCAST_1_3_5_11
LOC_1_ID_5_MCAST_2_3_4_9_12
LOC_2_ID_6_MCAST_2_7_8_9
LOC_2_ID_7_MCAST_6_8_10
LOC_2_ID_8_MCAST_6_7_9_10
LOC_2_ID_9_MCAST_5_6_8_10_16
LOC_2_ID_10_MCAST_7_8_9_17
LOC_3_ID_11_MCAST_4_12_13_14
LOC_3_ID_12_MCAST_5_11_13_15_16
LOC_3_ID_13_MCAST_11_12_14_15
LOC_3_ID_14_MCAST_11_13_15
LOC_3_ID_15_MCAST_12_13_14_19_21
LOC_4_ID_16_MCAST_9_12_17_18_19
LOC_4_ID_17_MCAST_10_16_18_20
LOC_4_ID_18_MCAST_16_17_19_20
LOC_4_ID_19_MCAST_15_16_18_20_21
LOC_4_ID_20_MCAST_17_18_29
LOC_4_ID_21_MCAST_15_19

Configuration for IP Test

LOC_1_ID_1_MCAST_2_3_4
LOC_1_ID_2_MCAST_1_3_5_6
LOC_1_ID_3_MCAST_1_2_4_5
LOC_1_ID_4_MCAST_1_3_5_11
LOC_1_ID_5_MCAST_2_3_4_9_12
LOC_1_ID_6_MCAST_2_7_8_9
LOC_1_ID_7_MCAST_6_8_10
LOC_1_ID_8_MCAST_6_7_9_10
LOC_1_ID_9_MCAST_5_6_8_10_16
LOC_1_ID_10_MCAST_7_8_9_17
LOC_1_ID_11_MCAST_4_12_13_14
LOC_1_ID_12_MCAST_5_11_13_15_16
LOC_1_ID_13_MCAST_11_12_14_15
LOC_1_ID_14_MCAST_11_13_15
LOC_1_ID_15_MCAST_12_13_14_19_21
LOC_1_ID_16_MCAST_9_12_17_18_19
LOC_1_ID_17_MCAST_10_16_18_20
LOC_1_ID_18_MCAST_16_17_19_20
LOC_1_ID_19_MCAST_15_16_18_20_21
LOC_1_ID_20_MCAST_17_18_29
LOC_1_ID_21_MCAST_15_19

References

[1] Bgp route aggregation. `http://packetlife.net/blog/2008/sep/19/bgp-route-aggregation-part-1/`. Accessed: 2019-04-01.

[2] RIPE NCC. Number of Remaining IPv4 Addresses. `https://labs.ripe.net/statistics/number-of-remaining-ipv4-addresses-daily`.

[3] Google. Ipv6 adoption. `https://www.google.com/intl/en/ipv6/statistics.html`.

[4] Brian E. Carpenter. Ip addresses considered harmful. *SIGCOMM Comput. Commun. Rev.*, 44(2):65–69, apr 2014.

[5] Ed D. Meyer, Ed. L. Zhang, and Ed K. Fall. Report from the IAB Workshop on Routing and Addressing. RFC 4984, RFC Editor, September 2007.

[6] R. Atkinson, S. Bhatti, and S. Hailes. Evolving the internet architecture through naming. *IEEE Journal on Selected Areas in Communications*, 28(8):1319–1325, October 2010.

[7] Saleem Bhatti, Ditchaphong Phoomikiattisak, and Bruce Simpson. Ip without ip addresses. pages 41–48, 11 2016.

[8] T. Cao-hoang and C. N. Duy. Environment monitoring system for agricultural application based on wireless sensor network. In *2017 Seventh International Conference on Information Science and Technology (ICIST)*, pages 99–102, April 2017.

[9] G. Panda and T. Saha. Building of low cost reliable wireless sensor network for smart indoor agriculture products. In *2018 2nd International Conference on Electronics, Materials Engineering Nano-Technology (IEMENTech)*, pages 1–5, May 2018.

[10] Delphine née Christin, Andreas Reinhardt, Parag S Mogre, and Ralf Steinmetz. Wireless sensor networks and the internet of things: Selected challenges. 01 2009.

[11] Ed T. Li. Recommendation for a Routing Architecture. Informational 6115, Internet Research Task Force (IRTF), February 2011.

[12] Nahla Abid. *Design of a user-level naming solution for the future Internet*. Theses, Télécom Bretagne ; Université de Rennes 1, January 2015.

[13] Z. Tang, Y. Zhou, W. Deng, and B. Wang. Lisp-hnm: Integrated fast host and network mobility control in lisp networks. In *2017 15th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pages 1–6, May 2017.

[14] Ditchaphong Phoomikiattisak and Saleem N. Bhatti. Network layer soft handoff for ip mobility. In *Proceedings of the 8th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless*

and Wired Networks, PM2HW2N '13, pages 13–20, New York, NY, USA, 2013. ACM.

[15] P. Chatterjee and N. Das. Multiple sink deployment in multi-hop wireless sensor networks to enhance lifetime. In *2015 Applications and Innovations in Mobile Computing (AIMoC)*, pages 48–54, Feb 2015.

[16] Noman Shabbir and Syed Rizwan Hassan. Routing protocols for wireless sensor networks (wsns). In Philip Sallis, editor, *Wireless Sensor Networks*, chapter 2. IntechOpen, Rijeka, 2017.

[17] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). Experimental 3626, Network Working Group, October 2003.

[18] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. Experimental 3561, Network Working Group, July 2003.

[19] D. Johnson, Y. Hu, and D. Maltz. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. RFC 4728, Network Working Group, February 2007.

[20] Jin-Man Kim and Jong-Wook Jang. Aodv based energy efficient routing protocol for maximum lifetime in manet. In *Advanced Int'l Conference on Telecommunications and Int'l Conference on Internet and Web Applications and Services (AICT-ICIW'06)*, pages 77–77, Feb 2006.

[21] Shijun He, Yanyan Dai, Ruyan Zhou, and Shiting Zhao. A clustering routing protocol for energy balance of wsn based on genetic clustering algorithm. *IERI Procedia*, 2:788 – 793, 2012. International Conference on Future Computer Supported Education, August 22- 23, 2012,Fraser Place Central - Seoul.

[22] M. Al-Rabayah and R. Malaney. A new scalable hybrid routing protocol for vanets. *IEEE Transactions on Vehicular Technology*, 61(6):2625–2635, July 2012.

[23] K. Beydoun and V. Felea. Wireless sensor networks routing over zones. In *SoftCOM 2010, 18th International Conference on Software, Telecommunications and Computer Networks*, pages 402–406, Sep. 2010.

[24] J. Lloret, M. Garcia, F. Boronat, and J. Tomas. A group-based protocol for large wireless ad-hoc and sensor networks. In *NOMS Workshops 2008 - IEEE Network Operations and Management Symposium Workshops*, pages 7–14, April 2008.

[25] Ospf implementation. `http://www.ciscopress.com/articles/article.asp?p=2294214`. Accessed: 2019-04-01.

[26] Mahesh K. Marina and Samir R. Das. Routing performance in the presence of unidirectional links in multihop wireless networks. In *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing*, MobiHoc '02, pages 12–23, New York, NY, USA, 2002. ACM.

[27] R. C. Shah and J. M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *2002 IEEE Wireless Communications and Networking Conference Record. WCNC 2002 (Cat. No.02TH8609)*, volume 1, pages 350–355 vol.1, March 2002.