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 also evaluates the effectiveness of the protocol along. The results showed that ...

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 #TODO NN,NNN 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

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 there 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.

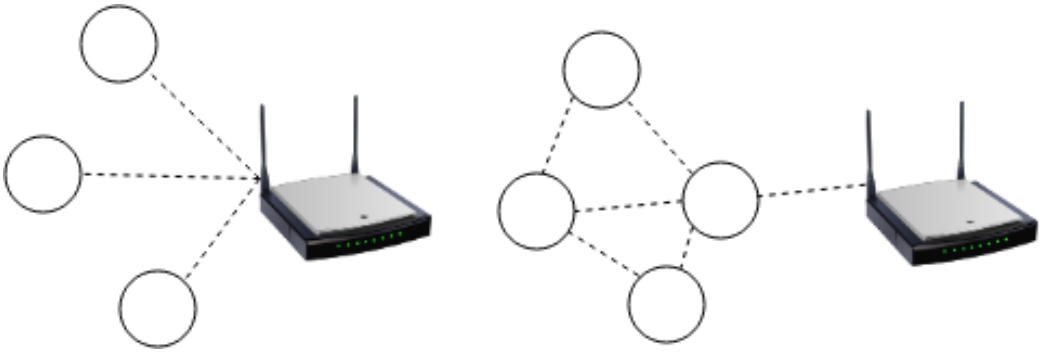

Figure 1: On the left are IoT devices connecting directly with a dedicated router, and on the right the IoT devices form a WSN in order to reach the router.

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.

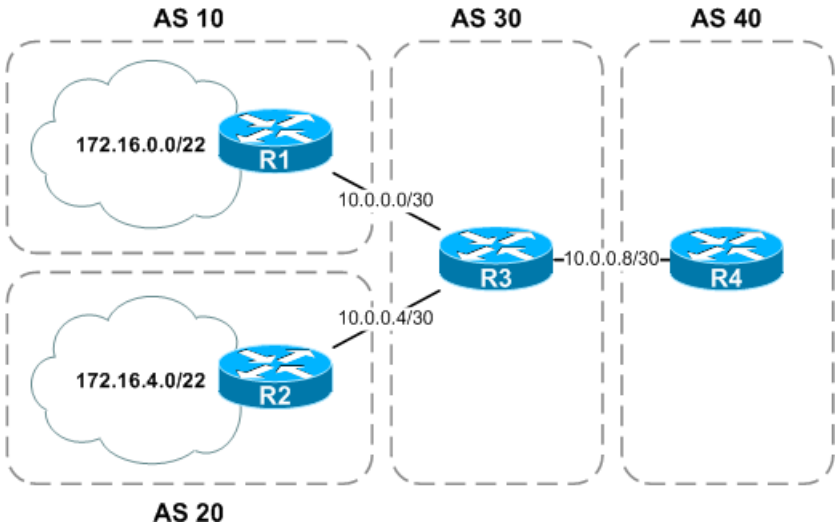

Figure 2: Example of address aggregation (Image from [1])

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 system 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 simply 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. 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 with smaller memory requirements for the RIB. The ID part of the address is unique to the host, and provides a fixed address which can be used at the transport layer.

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. ILNPv6 also requires some additions to the Domain Name System (DNS) to properly support multihoming.

The main goal of this project is to produce a library for a prototype communication protocol for 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. The native support for multihoming and multipath routing is especially beneficial to typically mobile internet of things (IoT) devices, and so this project aims to exemplify these benefits for agricultural sensor networks in particular.

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.

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.

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, and with ILNP the end destination identifier would remain the same, and so the underlying routing protocol would only have to include a way of realising a route to the new interface.

A different addressing scheme is only part of the solution required for successful intergration of WSNs and the internet. IoT devices typically have limited memory and power, and so require optimised routing protocols if they are to be integrated effeciently. 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 differeent routing protocols and provides a survey of protocols for each class. Routing protocols are either proactive, reactive, or hybrid.

2.1 Proactive

Each node maintains a routing table through knowledge sharing with adjacent nodes. Typically involves high overhead due to regular flooding and beacon messages, but performs better than reactive methods as mobility increases as links are repaired 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. Despite both making attempts to reduce enery consumption, they are defeated by reactive protocols in this

area.

2.2 Reactive

Nodes only seek out routes to remote nodes when one is required. Reactive protocols involve very little overhead in networks with low mobility, but performance degrades quickly as mobility increases. Examples include AODV [18] and DSR [19]. Since these protocols were designed for mobile ad-hoc networks, most research involves improving the protocols awareness of energy availability. When a single node lies on many paths, it can be quickly drained by traffic, and so [20] produced a solution that considered the mean energy of a path before choosing which route to suggest to the requesting node. Reactive protocols also perform well in wireless environments as lack of periodic updates reduces the chance of interference which exists with link state proactive protocols.

2.3 Hybrid

Features of both proactive and reactive protocols are used. The hybrid approach is useful in scenarios where the grouping of nodes is appropriate. [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 also use GPS to help route packets when their target is in a known location.

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.

Agricultural sensor networks often consist of many sensor nodes for monitoring the environment and a few actuators for triggering sprinklers or covering delicate crops. Based on this data, sensors 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.

Our ZBR 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 4 provides example communication paths, where data is communicated from the nodes in one zone to the sink in another zone.

The protocol was specifically designed for autonomous subnetworks to operate, while still being able to route globally. For example, sensors could communicate with actuators within their subnetworks, whilst sending monitoring data to a global sink. The only knowledge that would be available upon initialization is the ID and locator for the starting node, and the ID of the sink node. The sink node could then be multihomed without having to reconfigure the entire network of sensors. Sensors could also detect any other hosts within their subnetwork, which could be devices such as sprinkler systems, shelters, or cameras.

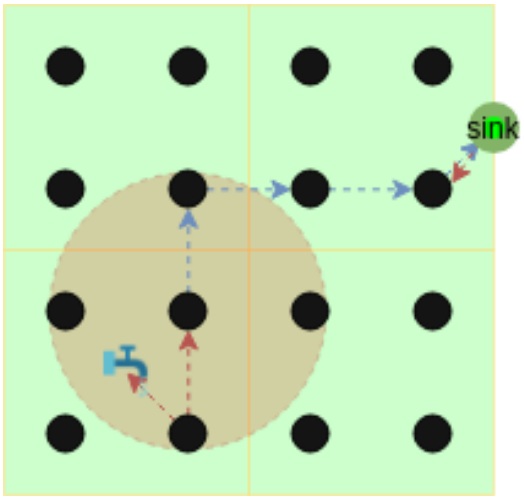

Figure 4: Communication patterns in agricultural WSN

A reactive protocol was chosen for communication between zones 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 zones.

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* message, containing its lambda value.

Neighbouring nodes reply with their current link state database (LSB) in a *LSBMessage*. This message contains the neighbouring nodes knowledge of the internal (within the locator) topology, as well as the links to external networks (other locators). Figure 5 gives a simple example of this communication. Whilst still in the intiliziation phases, neighbours will flood their entire LSBs to their neighbours, and once the messages they receive contain match their own LSBs, 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.

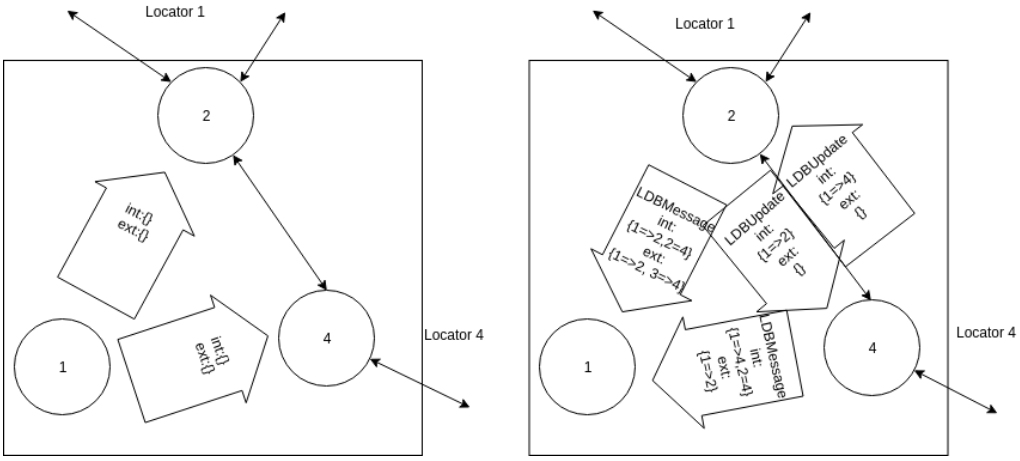

Figure 5: Exchange of LSB messages when a new node joins.

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 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.

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.

3.3 Inter-Zone Routing Protocol

AODV is a reactive protocol that is used differently for inter-zone and inter-zone routing. Inside locators, the hop list will consist of node IDs, providing exact routes between the source and destination. Between locators, the hop list will consist of locators, and the exact hop-by-hop routing will be provided by each zone internally.

AODV has three phases: discovery, maintenance, and recovery.

To discover routes to a given destination, AODV produces a list of hops that a packet can be sent over to reach a destination by flooding route request packets (RREQs). The route discovery process is summarised in figure 6, where the leftmost node (1) is requesting a route to the rightmost node (4).

Figure 7 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, which coupled with the source ID in the ILNP packet header can identify duplicate requests. Otherwise if this node's identifier already appears in the path so far, then it can also be discarded. These checks reduce unecessary duplication of request packets and avoid loops in the resulting paths.

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.

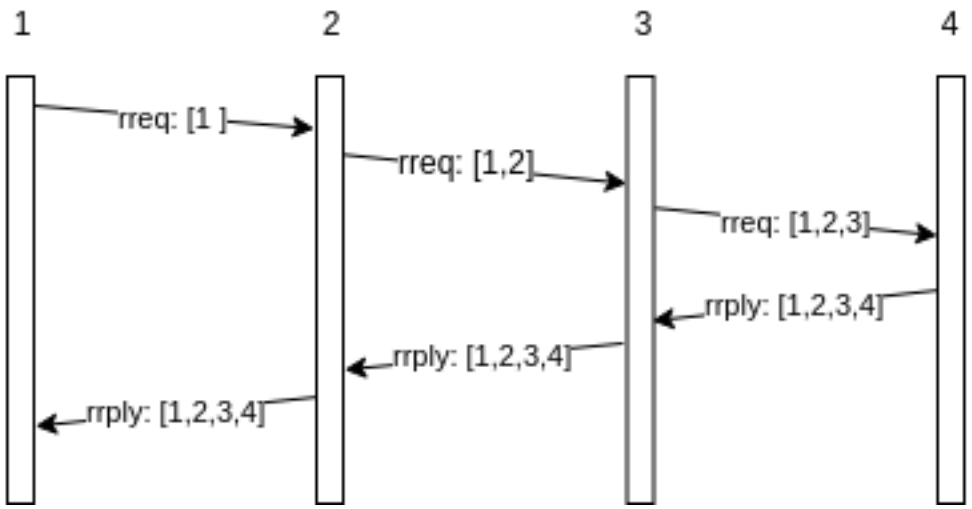

Figure 6: Route Discovery Overview

This provides robust communication if the end destination is multihomed (i.e. has interfaces to multiple locators), with less complexity than in IP. It also makes it easier to identify disjoint paths as the node has a single name in the network, which wouldn't be the case in IP multihoming.

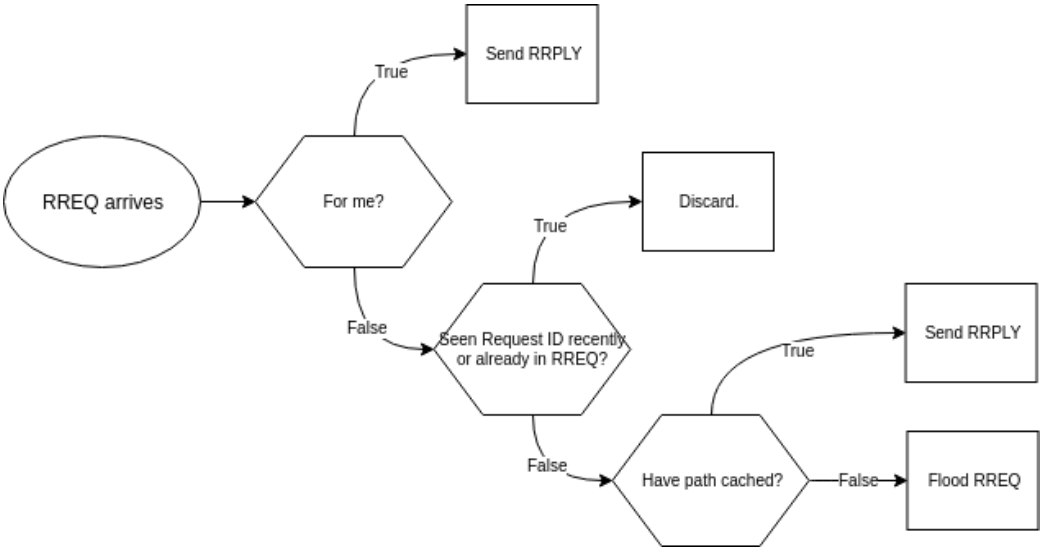

Figure 7: Route Request (RREQ) Flow

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 8 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 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. [24] 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.

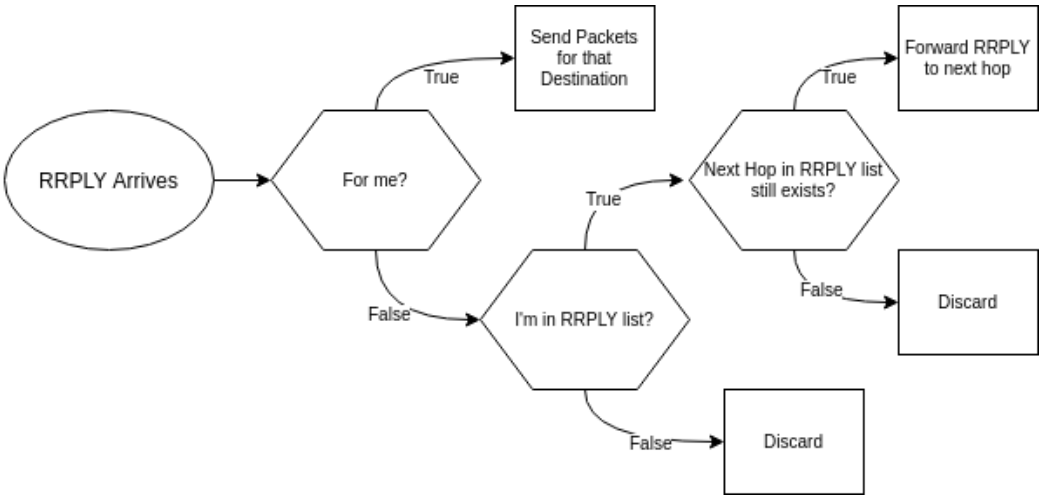

Figure 8: Route Reply (RRPLY) Flow

The structure of the route request is the same as in RFC3561 [18], but with an extra λ field in the header. This value provides a metric for determining the lifespan of a route, and is calculated for an individual node using equation 1, and visualised in Figure 9.

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

where L is the % of load the node is willing to give to networking, and E is the % battery remaining. It is based on the equation for λ in [25], 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, but to avoid network partions smaller values of E would have a greater effect on λ. Each node calculates this value before adding their ID to the path in the RREQ, and only changes the field in the packet if its calculated value is lower.

[25] also included a load balancing factor that would account for the number of neighbours a node had, and this was considered for use in the AODV implementation. Instead of using the number of neighbours as in their link state protocol, we could use the number of nodes using a route that passed through this node. This value could be ascertained by counting the number of unique source IDs of packets that were forwardded. However, defining a maximum in order to normalise this factor would place restrictions on how large the network could grow. Too large and the factor has no real effect, too small and it would reduce to zero making the metric meaningless for nodes further upstream. If this number could be supplied however, it would help other nodes choose routes that aren't being used as heavily, which would be very beneficial for the lifespan of the network.

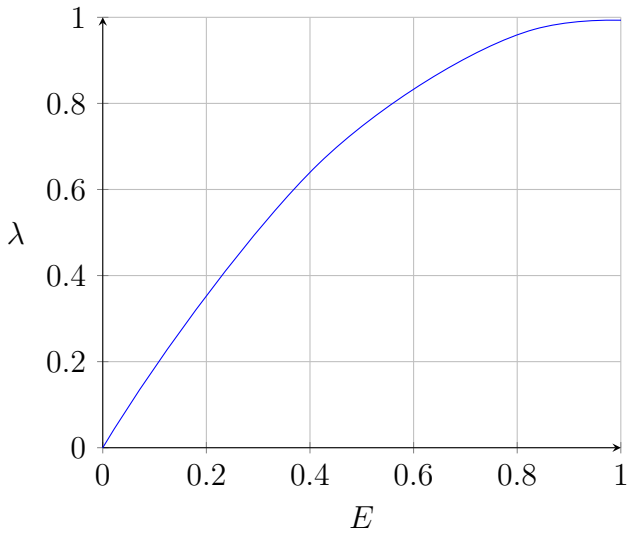

Figure 9: Plot of Equation 1

In order to check if neighbours are still live, *HELLO* messages are exchanged at regular intervals between neighbouring nodes. These messages also provide neighbour discovery as they are limited to one hop, therefore the source ID can be mapped to the arriving IPv6 address in the emulated link layer.

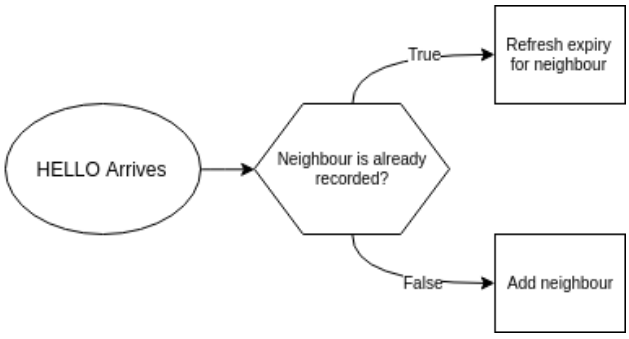

Figure 10: Hello Flow

Figure 10 shows how the *HELLO* messages are processed. If a node fails to deliver a hello messages after a certain interval, then it's neighbours will begin the route recovery process.

Cached routes will also expire after some time to trigger another route request if required. This allows nodes to learn if a path is under heavy load, and ensures that a route will be corrected if for some reason the recovery process is not completed.

If the keepalive process fails for a neighbour, then a node will forward a RERR to all previous nodes in any paths it takes part in, shown in figure 11.

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.

The intra-zone routing table is responsible for forwarding packets between the nodes of a locator. To do this, it keeps track of:

1. The destination node ID
2. The next hop ID to reach the destination ID
3. The cost of this path, based on λ.

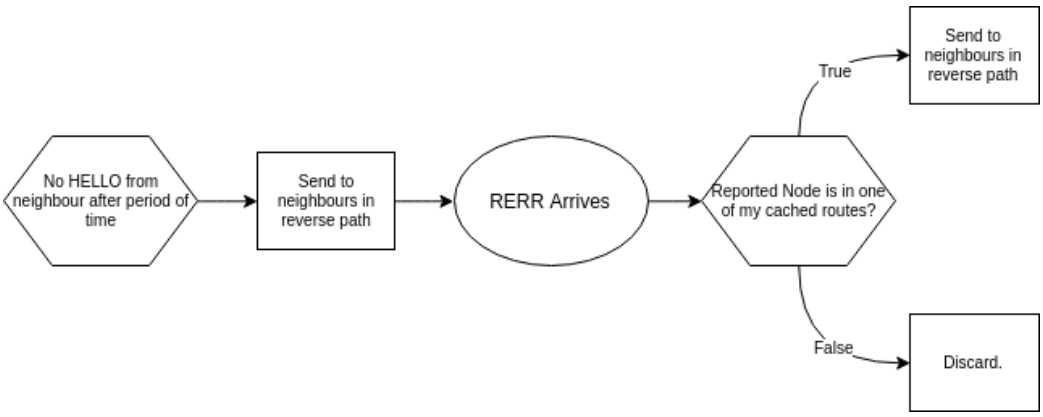

Figure 11: Route Error (RERR) Flow

4. Whether or not the destination node is a border or normal node.

5. The neighbouring locators that the destination node has links to (border nodes only).

The intra-zone routing table is built by broadcasting RREQs where each hop is an node ID, and only contains routes to actuator nodes and border nodes.

The inter-zone routing table is responsible for forwarding packets between nodes in different locators, and is maintained by nodes that exist on the borders of these locators such as those highlighted red in figure 12. It contains information about:

1. The destination locator

2. The next hop locator to reach this destination (i.e the ID of a border node from a neighbouring locator)

If multiple border nodes provide links to the same neighbour zone, then nodes 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.

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;

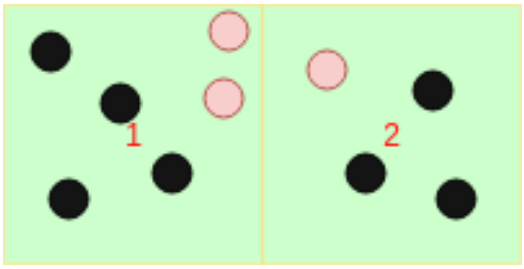

Figure 12: Border Nodes

Figure 13 shows 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.

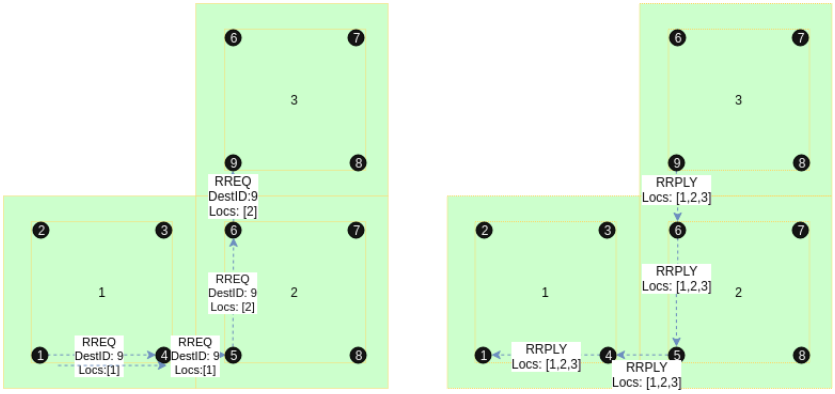

Figure 13: Locator Discovery

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 14 shows the components with the flow of messages.

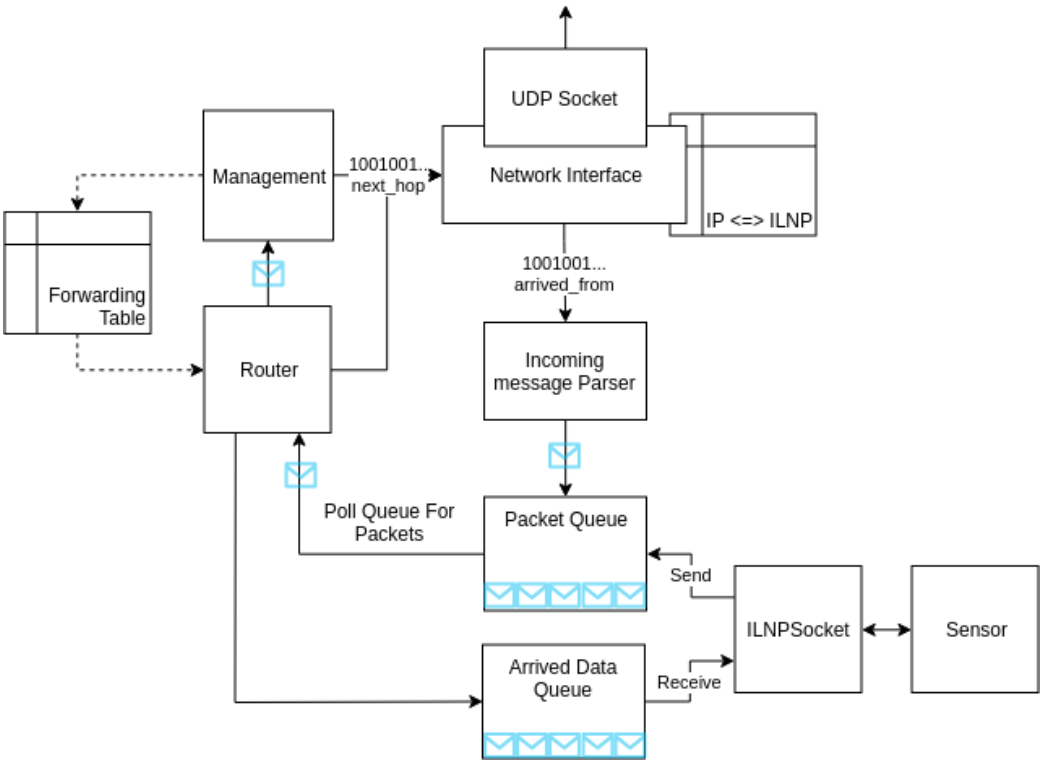

Figure 14: Implementation Structure

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 broad-

casted packets. Figure 15 shows the ranges of each sensors radio signal as dashed lines, where each node within the range joins the multicast group.

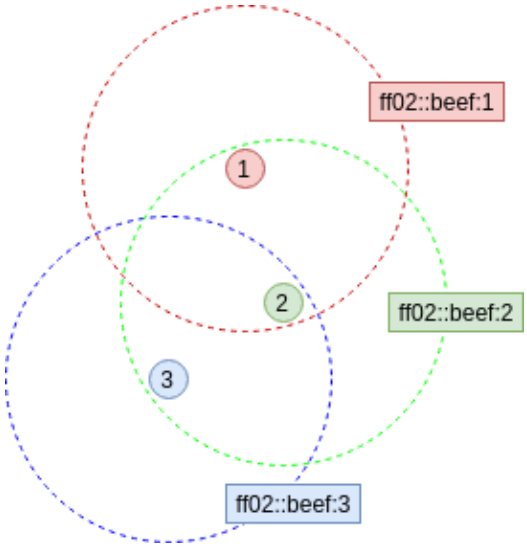

Figure 15: Wireless broadcast emulation using multicast

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 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.

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.

The router polls the incoming packet queue and decides the next course of action.

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.

#https://ieeexplore.ieee.org/document/7993954 to fix route reply storm issue: add jitter before responding with route reply and listen for other replies

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].

Only the ID of the sink is known by each node, which is all that should be necessary when using ILNP to route a packet. If it was possible to provide a FQDN for the sink node then the addressing scheme would be abstracted also, but emulating a DNS was deemed unecessary for the experiment.

5 Experiment

The aims of the experiment were to show that protocol succeeded in meeting the goals from section **??**, and to exemplify how ILNP helped in achieving these goals compared to how it would be implemented in IP. This was be achieved by simulating a wireless sensor network for monitoring fields in a farm.

Figure 16 shows the real life scenario being emulated, where each black circle represents a sensor node, and the green circles represent interfaces to the sink, such as be wireless receivers. The transparent red circles represent the wireless ranges of the nodes, and the blue dotted arrows show an example path that could be taken to reach the sink.

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[26])
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. The hypothesis being that the protocol will perform better using ILNP since nodes will be able to operate without knowledge of the entire network. Figure 17 shows that ILNP can emulate IP by either:

1. providing a locator for every node, essentially treating every node as a subnetwork that will require a routing table entry.
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, so either could be used to the same effect. The locator per node approach was taken for this experiment.

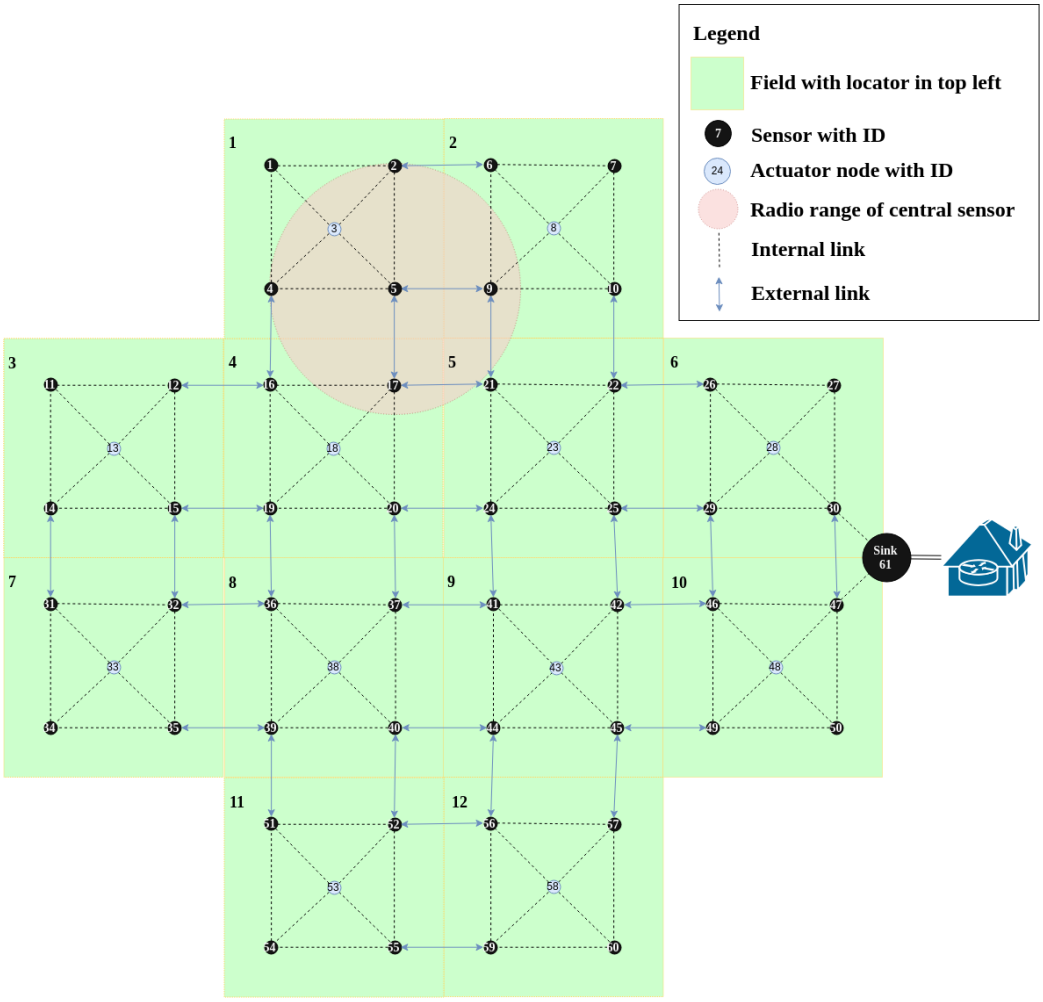

Figure 16: WSN layout with example routes.

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 using a list of IP addresses of the lab machines in order to know which machines were available for use.

Since multicast was being used to emulate wireless broadcast, my user ID

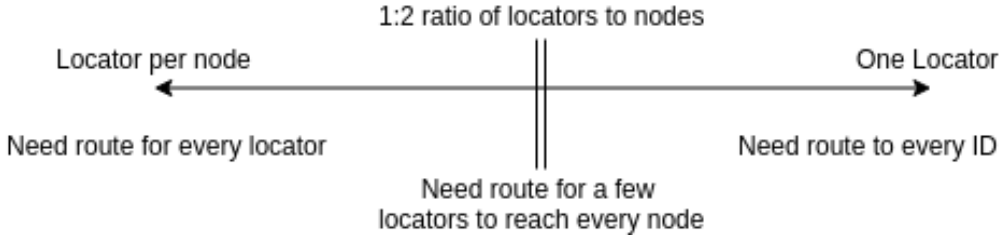

Figure 17: Relationship between IP and ILNP

was used as the second-last octet to avoid collision with other experiments possibly running on the network.

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. The monitor would also report the size of the routing table when each packet was sent.

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, which

included a sequence number so that the level of disordering and packet loss that occurred could also be measured. Loss would simply be the difference between the number of data packets sent by each sensor and the number received by the sink. Disorder was measured as the absolute different between the current highest sequence number received from a node (i.e the sequence number in the most recently sent packet) and the sequence number in the received packet.

A *battery* module was also added to the emulation. The battery would be decremented as each packet was sent, emulating real loss of power when transmitting which would be used when calculating λ for each node.

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

5.3 Results

Figures 18 and 19 show the concentration of data and control packets throughout the network during the intialisation of the network in the multiple sink scenario. As expected, the overhead is initially quite high, as every node is trying to learn routes to the sink.

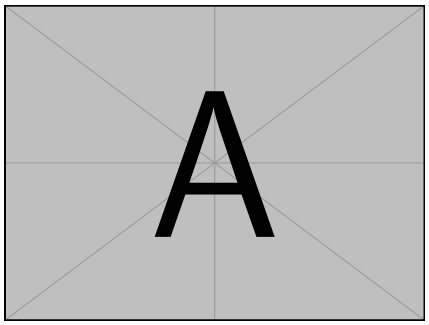

Figure 18: Data packets sent by each node during network initialisation.

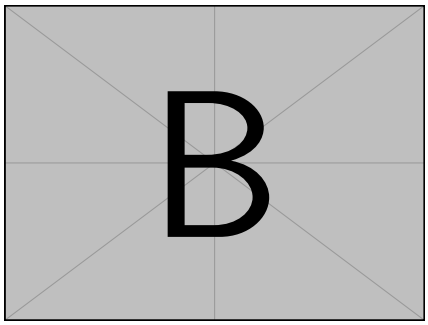

Figure 19: Control packets sent by each node during network initialisation.

Figures 20 and 21 show how the overhead decreases once routes are established, and how the paths taken are fairly evenly spread and not just focused around the shortest paths.

Figures 22 and 23 show how as nodes begin to fail, the limited number of paths force heavier traffic to pass through the remaining nodes. The failing

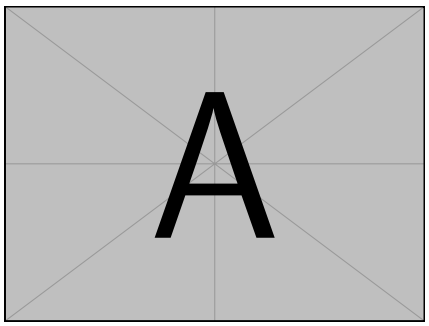

Figure 20: Data packets sent by each node once initial routes have been established.

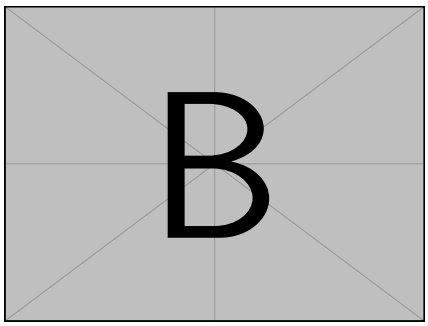

Figure 21: Control packets sent by each node once intial routes have been established.

nodes have occurred near the sinks as expected.

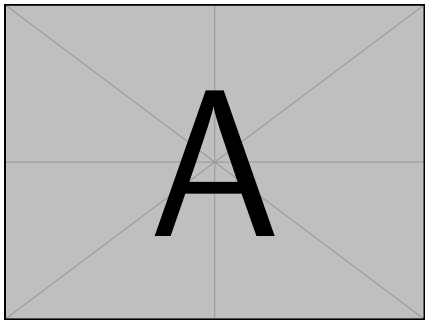

Figure 22: Data packets sent by each node once failures have started to occur.

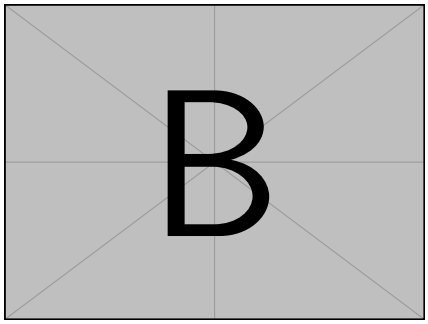

Figure 23: Control packets sent by each node once failures have started to occur.

Figure 24 shows the packet over time, and finally figure 25 shows the level of disordering over time.

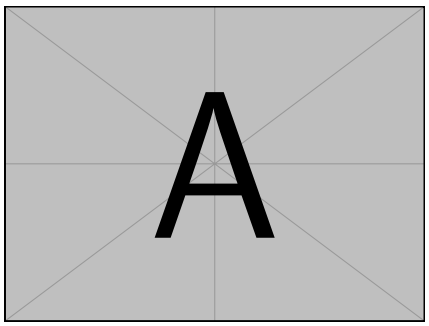

Figure 24: Packet loss over time.

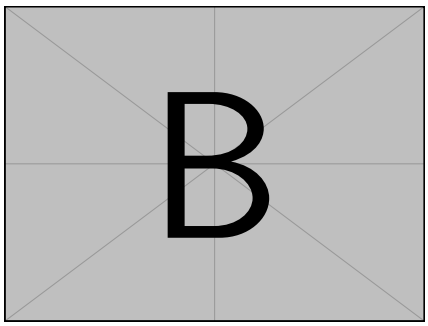

Figure 25: Disordering over time.

Figure 26 shows the sizes of the internal forwarding tables for IP and ILNP, and 27 shows the sizes of the external forwarding tables.

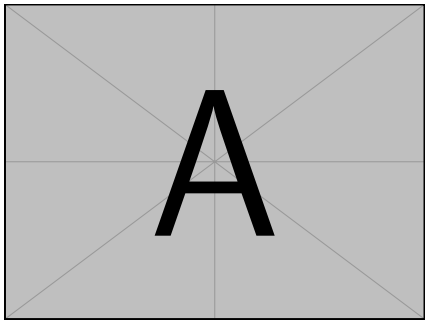

Figure 26: Sizes of internal forwarding tables over time.

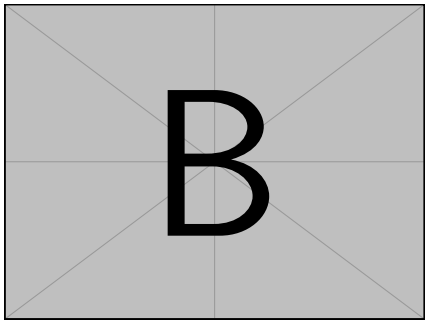

Figure 27: Sizes of external forwarding tables over time.

6 Discussion

1. Explain features of heat map and snapshots
2. Discuss how well the routing protocol performed, compared to other implementations.
3. Discuss weaknesses with experiment

7 Conclusions

Discuss how global AODV was originally considered, but discarded since it didn't take full advantage of ILNP.

Possible mercy message to send from nodes under heavy load to try and request downstream nodes to use different paths, rather than outright route errors.

was the goal met, and if so how well?

future work with ILNP, possible suggestions of better alternatives to the routing protocol used.

8 Appendix

1. Instructions on installing, and executing and using the python module

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] 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.

[25] 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.

[26] 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.