Supporting Device Mobility and State Distribution through Indirection, Topological Isomorphism and Evolutionary Algorithms

Andrew Attwood

A THESIS SUBMITTED IN PARTIAL FULFILMENT
OF THE REQUIREMENTS OF LIVERPOOL
JOHN MOORES UNIVERSITY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

July 2014

Abstract

The Internet of Things will result in the deployment of many billions of wireless embedded systems, creating interactive pervasive environments. These pervasive networks will provide seamless access to sensor actuators, enabling organisations and individuals to control and monitor their environment. The majority of devices attached to the Internet of Things will be static. However, it is anticipated that with the advent of body and vehicular networks, we will see many mobile Internet of Things Devices. During emergency situations, the flow of data across the Internet of Things may be disrupted, giving rise to a requirement for machine-to-machine interaction within the remaining environment.

Current approaches to routing on the Internet and wireless sensor networks fail to address the requirements of mobility, isolated operation during failure or deal with the imbalance caused by either initial or failing topologies when applying geographic coordinate-based peer-to-peer storage mechanisms. The use of global and local DHT mechanisms to facilitate improved reachability and data redundancy are explored in this thesis. Resulting in the development of an Architecture to support the global reachability of static and mobile Internet of Things Devices. This is achieved through the development of a global indirection mechanism supporting position relative wireless environments. To support the distribution and preservation of device state within the wireless domain a new geospatial keying mechanism is presented, this enables a device to persist state within an overlay with certain guarantees as to its survival. The guarantees relating to geospatial storage rely on the balanced allocation of distributed information. This thesis details a mechanism to balance the address space utilising evolutionary techniques. Following the generation of an initial balanced topology we present a protocol that applies Topological Isomorphism to provide the continued balancing and reachability of data following partial network failure.

This dissertation details the analysis of the proposed protocols and their evaluation through simulation. The results show that our proposed Architecture operates within the capabilities of the devices that operate in this space. The evaluation of Geospatial Keying within the wireless domain showed that the mechanism presented provides better device state preservation than would be found in the random placement exhibited by the storage of state in overlay DHT schemes. Experiments confirm device storage imbalance when using geographic routing; however, the results provided in this thesis show that the use of genetic algorithms can provide an improved identity assignment through the application of alternating fitness between reachability and ideal key displacement. This topology, as is commonly found in geographical routing, was susceptible to imbalance following device failure. The use of topological isomorphism provided an improvement over existing geographical routing protocols to counteract the reachability and imbalance caused by failure.

Contents

List of Figures

List of Equations

List of Tables

List of Acronyms

6LowPAN	IPV6 for Low Power Networks
AODV	Adaptive On Demand Distance Vector Routing
AODV	Adaptive On demand Distance Vector routing
ASOS	Ad Hoc Storage Overlay System for MANETS
BGP	Border Gateway Protocol
COA	Care of Address
CoAP	Constrained Application Protocol
CoRE	Constrained RESTful Environments
DAG	Directed Acyclic Graphs
DHT	Distributed Hash Table
DODAG	Direction Orientated Directed Graph
DS	Distributed System
DSDV	Destination Sequenced Distance Vector Routing
DSR	Dynamic Source Routing
DSSS	Direct Sequence Spread Spectrum
EBL GHT	Evolve Balance Localise GHT
EDCA	Enhanced Distributed Channel Access
ER	Edge Router
ESS	Extended Service Set
ETSI	European Telecommunications Standards Institute
FFD	Fully Functional Device
FMIP	Fast Handovers
GA	Genetic Algorithm
GHT	Geographic Hash Table
GPS	Global Positioning System
GPSR	Greedy Perimeter Stateless Routing
HIP	Host Identity Protocol
HIT	Host Identifier Tag
HRA	Hierarchical Routing Architecture
HTTP	Hyper Text Transfer Protocol
HWMP	Hybrid Wireless Mesh Routing Protocol
HWMP	Hybrid Wireless Mesh Routing Protocol
IaaS	Infrastructure as a Service
ID	Identity
IDRM	Inter Domain Routing for MANETS
IEEE	Institute of Electrical and Electronic Engineers
IETF	Internet Engineering Task Force
IETF	Internet Engineering Task Force
IIA	Indirection Identity Address
IoMANETS	Indirection Overlay for MANETS
IoT	Internet of Things
IP	Internet Protocol
IPv6	Internet Protocol Version 6
ISO	International Standards Organisation
ISP	Internet Service Provider
ISRP	Iterative Successor Pointer Rewiring Protocol

LEACH	Low Energy Adaptive Clustering Hierarchy
Low PAN	Low Data Rate Personal Area Networks
M2M	Machine to Machine
MAC	Media Access Control
mAH	milliampere-hour
MANET	Mesh Ad-Hoc Network
MAP	Mobile Access Point
MD5	Message Digest 5
MDA	Mesh Deterministic Access
MDT	Mobility Border Protocol Domain Tree
MoSIPD	Mobility Overlay Supporting Indirection and Packet Delivery
MP	Mesh Point
MPLS	Multi-Protocol Label Switching
MPP	Mesh Portal Point
MSDU	Mac Service Data Units
NAT	Network Address Translation
NAV	Network Allocation Vector
NEMO	Network Mobility
NEMO	Network Mobility Basic Support Protocol
OLSR	Optimised Link State Routing Protocol
OSI	Open Systems Interconnection Model
P2P	Peer to Peer
PA-FMIP	Past Association FMIP
PAN	Personal Area Network
PLC	Programmable Logic Controllers
PR-DHT	Position Relative DHT
QOS	Quality of Service
RAM	Random Access Memory
RANN	Route Announcements
RFD	Reduced Functional Device
ROLL	Routing Over Lossy Links
ROM	Read Only Memory
RPL	Routing Protocol for Lossy Links
RREQ	Route Request
RSSI	Received Signal Strength Indicator
SAN	Sensor Actuator Networks
SCADA	Supervisory Control and Data Acquisition
SCCIR	Smart Cities Critical Infrastructure Response
SOAP	Simple Object Access Protocol
SSID	Secure Service Identifier
SWRL	Semantic Web Rule Language
TC	Topology Control
TEEN	Threshold sensitive Energy Efficient sensor Network protocol
TIR	Topological Isomorphic Routing Protocol
TOR	The Onion Router
TSAP	Transport Service Access Point
UDP	User Datagram Protocol
URI	Universal Resource Indicator
VRR	Virtual Ring Routing
WAN	Wide Area Network

WDS	Wireless Distribution system
WIFI	Wireless Fidelity
WMBP	Wireless Mobility Border Protocol

Acknowledgements

I would like to thank my supervisor, Dr Omar Abuelmaatti, for his support and guidance during my research.

I would especially like to thank my family and friends for their encouragement and support.

Chapter 1

Introduction

The Internet has become an essential element of society: it provides digital substrata, pervading every aspect of our lives. Devices attached to the Internet produce and store 'Big Data'; this, combined with powerful analytics, provides the information and decision-making capabilities that enable our economy, personalises healthcare and adapts our personal environments. Advances in technology are further extending the pervasiveness of this network. The Internet, as it is used today, can be seen as providing information about things, whereas the future Internet will provide communication between things. This is commonly referred to as 'The Internet of Things'—a vision of an Internet with many billions of devices engaged in Machine-to-Machine (M2M) communication.

Achieving this step change will require a re-engineering of the current Internet—the start of which can already be seen with the introduction of Internet Protocol version 6. The change in addressing scheme is due, in part, to the predicted increase in Things devices being connected to the Internet; the increase in address space afforded by IPv6 will enable Internet of Things devices to have identity without the need for masquerading services that restrict the global reachability of Internet connected devices. These changes will enable greater innovation through connectivity.

There have been a number of technological advances that have changed our interaction with the Internet. One of the greatest changes is associated with mobile computing; this has provided the ability for people to share information, through social networking and other applications, without the restriction of being in a specific location. Users are no longer bound to the home or office; rather, users can connect to the Internet and share information whenever they require. These advances have reduced the barriers between our physical and digital worlds. Social acceptance of this technology has led to increasing demands for innovation to keep pace with user requirements. The expectation that devices are 'connected' is now commonplace, with most home technology, such as TV's, house alarms, radios, etc., being sold with wireless technology enabled. These are the first wave of devices to be connected; they are not limited by power

consumption and normally operate using 802.11 wireless technologies, thus providing fast communication and a level of security. This type of connectivity is relatively well established, with most homes equipped with access points connected to the Internet. In order to create pervasive embedded environments, smaller micro devices will need to be deployed to provide the level of coverage and sensing capability that is required of future use cases. The use of wireless sensor technology in the home has been restricted largely by a lack of standardisation. However, advances in standardisation and the level of technological advancements are moving forward, making a pervasive Internet a reality. Processor, storage, wireless and battery technology have advanced to a point where embedded devices are capable of the processing capabilities and bit rates required to carry and process Internet traffic. The advances in technology coupled with the work that is being carried out, creating Internet of Things protocols, means the Future Internet of Things will soon be a reality.

Wireless sensor network technology has long been seen as an enabling technology for a ubiquitous smarter planet, with the capability to perform low-cost environmental monitoring utilised for many years by the research community. Initially, it was thought that the capability of these devices would limit their interaction with the wider internet; instead, they were typically co-ordinated using proprietary protocols, limiting the interoperability of software and protocol stacks. However, in recent years, due to the rising capability of embedded technology, the introduction of IP to low-power wireless devices has been achieved: it is now possible to connected devices to the Internet that will be active for many years on a single battery; these devices are now able to process IP communications with distant correspondents on the Internet.

The increased fidelity provided by the mass deployment of these sensing devices will enable us to better manage our homes, offices and industrial premises, and will provide many additional benefits aside from the device's primary roll. For example, if you had a house fire, the devices could help first responders locate you and provide assistance. For industrial settings, devices have the propensity to identify the presence of harmful substances and alert operators or alternatively to directly apply Machine-to-Machine communication to a device that can help mitigate the problem. This changes slightly the nature of the devices interaction with the Internet because now it becomes essential that a device is able to propagate the information it holds to others. In certain scenarios, it could mean the difference between life and death if devices are unable to communicate, which is especially important when dealing with Critical Infrastructures.

Critical infrastructures play an important role in ensuring the wellbeing of the populace. Protecting critical infrastructures and ensuring their continued operation will be an important part of the Future Internet. Equally important is the set-up of systems relationships of which a failing system is a part as, under certain circumstances, these could render a minor system critical. Infrastructure failure is usually brought under control through system adaptation, such as using sensor area networks to close valves or by emergency response, e.g. extinguishing fires. Current response procedures rely on antiquated information-sharing techniques, and provide little or no opportunity to effect change within the failing Infrastructures systems. There may also be minimal understanding surrounding the important systems of systems' roles being provided by individual components of the failing system. This will require new ways of sharing information with those that are responsible for responding to incidents.

The Internet of Things will provide new forensic capability by deploying thousands of sensing devices to monitor areas and providing them with the ability to form new ad-hoc relationships. Devices will eventually hold valuable information to assist investigators; this could be following an incident or so as to facilitate access to historical records to identify patterns in behaviour. Following incidents, devices will hold information essential to discovering cause, which will provide new insight and aid the evidence-gathering capabilities of authorities and organisations. Organisations would benefit from identifying the root cause of industrial failure. Alternatively, the police or fire service could identify the actions preceding domestic events, such as house fires, burglaries, etc. In some cases, this would need to be accomplished in a situation that a smart space enabled with sensors might be partially or totally destroyed. The capability for devices to work cooperatively to ensure the collective memory of the system is a key requirement in the Internet of Things. This is viewed as a type of distributed memory, formed through ad-hoc relationships formed between low-power systems.

The levels of assurance required from the devices that protect the wellbeing of individuals are high. Devices will be expected to work even during times of difficulty, with such requirements seeing the levels of fault tolerance found in Industrial applications of Internet of Things devices being applied to the home. These requirements will dictate developments in this area, requiring that devices are able to communicate in the face of failure—not only with owners of the infrastructure, but with those who require it, necessitating adaptive scalable solutions to emerging Internet of Things situations. This is challenging considering the low-power nature of the devices involved, as well as their limited connectivity.

The Internet was originally designed to provide services to machines that are fixed to a specific point of attachment. These machines were not typically restricted by the resources they consume; their capabilities, as well as their connectivity, experienced growth in line with user requirements. Servers and workstations are now equipped with high-speed processors interconnected at high data rates, running a range of Internet-enabled applications. A plethora of routing protocols support the Internet's distributed applications via the converged global address space. It is this converged address space that provides us with the end-to-end connectivity that those distributed applications require. The physical connectivity provides bandwidth advantages but restricts a device from forming direct relationships with other networks. Wireless sensors are the opposite in this regard, having limited bandwidth whilst being capable of forming new ad-hoc relationships.

The Internet does provide the opportunity for devices to form other virtual topologies: for example, the TOR network and P2P download services. These overlay protocols provide fault tolerance through distribution, enabling services to operate in the face of adversity. Similar approaches should be applied to the Internet of Things to provide the resiliency required.

The next generation Internet will result in a topology that is quite different to its current form, with many more devices being connected to the wireless space beyond the physically connected network. The advancement of mobile telephones is a starting point; eventually, IoT devices will

be embedded into medical devices, bikes, clothing, and disposable items, such as food packaging. Static devices will be embedded into items facilitating advanced interaction, with typical devices including vending machines, information panels, automated check-in systems and home appliances. Pervasive requirements will also see the implementation of distributed applications that monitor our actions, using information obtained from devices forming the Internet of Things. These applications include health, transport, safety and logistics. Mobile and static devices must interconnect to provide the fidelity required by the distributed applications and also to support the autonomous Machine-to-Machine interactions.

The Internet of Things will utilise wireless Mesh technology so as to provide coverage between the nodes that make up the ubiquitous environments, providing connectivity back to a point of connectivity with the Internet. Individual nodes will form collectives that support one another to meet the goals of the system. As an example, a primary goal of the system is to establish and maintain communication between the devices that constitute the network, external networks and the Internet. In order to realise this, Wireless Mesh Networks must exhibit fault-tolerant behaviour.

Mobility support within the Internet of Things will be essential. Devices will often be mobile in groups, and will come into contact with other mobile groups or different edge networks that are not part of the device's same home network. This will ultimately cause reachability issues, with the mobile node unable to communicate with its home network or other correspondents on the Internet. There has been a great deal of research directed towards the mobility of devices and networks, with these schemes not widely adopted, with devices instead commonly relying on multiple accounts and the allocation of temporary addresses to obtain connectivity between devices and their home networks. In an effort to realise the full potential of the Internet of Things, improvements to existing protocols are needed to support the reachability of low-power devices whilst being mobile and providing a mechanism for greater Machine-to-Machine interaction.

1.1 Motivation

With the advancement of technology, homes and businesses can be made safer for those who use them. The Internet of Things will provide humanity with the opportunity to make our environments more able to react and respond to situations that would normally cause the loss of life or cause serious damage to infrastructure. Due to their low cost and ease of installation, the Internet of Things will be constituent to many billions of individual wireless sensors, which will result in any individual deployment having as many as several hundred—if not many thousands—of individual devices.

The richness of the data provided by these devices will help systems to respond quicker to situations and accordingly permit those responding to do so with greater insight, reducing or mitigating the impact of the failure. Devices with Sensor Actuator capability will be able to

intervene within their own environment, either through autonomic operation or through the facilitation of first-responder intervention. Providing facilities beyond monitoring will enable first responders to do more to help alleviate potentially dangerous situations. It is envisaged that people, through the deployment of Body Area Networks will also be capable of directly interacting with the Internet of Things, which provides an additional benefit to first responders as they can utilise this information to locate and identify the status of individuals using ad-hoc communication between the Internet of Things devices situated in the failing environment, responder networks and the Body Area Networks of those involved. This combination of information will provide all involved with the best opportunities to both respond and survive.

It is important that information generated by sensors leading up to, during and following incidents is saved as post-failure analysis is an important aspect of future critical infrastructure protection. Providing the capability to learn from failure at the level provided by the Internet of Things will provide invaluable insight, facilitating the opportunity to learn so as to mitigate future failure. Providing the capability for devices to save their state in a distributed way is essential, as during the failure it would be expected that a proportion of the devices that make up the local sensor network would fail permanently; there would also be the possibility that devices experience periods without Internet connectivity, thus rendering them unable to transmit information to Internet-based control systems.

The ability for devices to self-organise and respond to situations will enable them to operate for longer in the face of failure. This ability to self-organise is quite unique and inherent to the transient nature of wireless communications. Fixed networks are limited in their response to failure: they can change their view of the world to deal with distant faults; however, if the failure is local this usually results in isolation. Wireless networks have the ability to form new relationships in an ad-hoc way. The ability for a first responder to interact with a failing infrastructure is quite unique to wireless Mesh networks.

The exploration of wireless sensor networks acting as a distributed data structure is central to this work. Providing wireless systems with the capability to identify their location and create distributed data spaces that are tolerant to failure will provide many benefits, including not only a richness of information during times of optimal operation but one that has the capability to adapt, both to protect itself (distributed state) and the users of the system and infrastructure through improved incident response.

The specific research motivations for this thesis are to address the issues of device reachability when those devices are both mobile and in contact with the internet. The proposed solutions will need to address the issues inherent to existing approaches when dealing with the types of devices that exist in the IoT e.g. managing the sleep state of devices. Existing schemes operate under the assumption that mobile nodes are associated with a single home location that is always available. This will involve extending existing work relating to mobility protocols for mesh ad hoc networks, specifically the extension of indirection architectures to support wireless mesh networks. Indirection mechanisms can be extended to counteract the intermittent nature of device connectivity. Global indirection has the potential to support the mobility of devices but to maintain the capability of systems to preserve individual device state, given the failure of a

proportion of the wireless mesh network, will require a local mechanism to support the device or devices when they become isolated. To provide this capability, this thesis will resolve the issues of distributed storage within the wireless domain. This will be achieved through the use of DHT within the wireless domain, enabling devices to maintain information despite the loss of a proportion of the network. Constructing DHT or position relative DHT referred to as GHT can be complex and rely on specialist hardware units e.g. GPS. Utilising GHT for storage as proposed in this thesis, will introduce balancing concerns. Addressing the balancing of distributed state amongst devices to facilitate a fairer distribution of work load and reduce the impact of the failure of individual devices will be paramount. To facilitate this we will investigate the use of localisation approaches and look to extend the existing work to provide balanced identity spaces for wireless GHT. Though the application of Genetic Algorithms we will evolve suitable topologies for GHT. The use of Genetic Algorithms will require experimentation to identify methods for evaluating the fitness of a proposed candidate topology. This will require multiple variable fitness functions to balance the various topological requirements. The solution driven by the context will require that the topology generated will be capable of being adapted to mitigate the impact of failure. This will result in the development of a greedy forwarding routing protocol for GHT to best balance the topology given the constraints of the environment following failure.

1.2 Problem Definition

Existing communication protocols lack the capability to provide the minimum required service that will be expected of the future Internet of Things. This is especially true when reviewing the requirements of critical infrastructure protection and the capability of devices to persist knowledge about the state of a system whilst it is undergoing the transformation from a working to failed state.

It is essential that Internet of Things Devices are capable of operating in environments where they are attached to the Internet and working in conjunction with higher-level platforms, such as home energy management systems and Future City co-ordination systems. It is also essential that they can continue to operate when that connection is removed and the device becomes isolated from the wider Internet, and all that remains being the local connectivity to other wireless devices in the same Wireless Mesh Network. Devices must be capable of operating in both a coordinated manner whilst, at times, being capable of self-organising to overcome adversity. This capability will provide smart spaces with the capacity to reorganise, providing the capability to overcome faults that occur or at least reduce their impact. Typically, systems have a primary goal, e.g. House Alarm Panel. They are limited by their connectivity and capability to support just this single goal. This will not be possible in an Internet of Things where individuals and organisations alike become reliant on devices to provide information about the world in which they interact to services beyond that of the primary goal.

Wireless sensor networks are a type of distributed system and, as such, individual devices must play their part to achieve the goal of the collective. Devices must also be capable of operating independently, as through failure or placement devices might be separated from the wider Internet of Things for extended periods. When connected, devices form smart spaces (a subset of the Internet of Things) must be capable of cooperating to effectively distribute processing and storage load. A benefit of bringing Internet connectivity to sensors is the provision of systems capable of performing 'Heavy Lifting', meaning devices must harness the power of those systems that are capable of providing additional processing resource and insight to save their own limited resources. The first problem in this regard is related to network topology: how does one organise a distributed wireless system so data can be routed to and from low-power devices in an efficient way? Moreover, when that network is connected to the Internet, how does a network of low-power devices cooperate to facilitate global reachability of devices and things? For a sensor, network fairness is important when all nodes are equal. In a truly distributed system, all devices are equal—perhaps not in capability but certainly in their importance as a key requirement of a distributed system—where information and system overhead (routing) is shared equally amongst all nodes in the system. If the Internet of Things is considered a distributed system comprising devices that are highly capable and those that are highly embedded, this leads to an imbalance as the storage should be distributed to the Internet, with sensors only resorting to storage if the situation requires.

For the purpose of this study, this thesis considers a system as having two states. The first is during normal operation, where the system should minimise its overhead to prolong network life. This would involve the wireless systems transmitting information infrequently to each other and to a base station to report on their observed environmental conditions. The network should also be receptive to mobile devices or collections of devices forming associations as the Future Internet will be one of ad-hoc mobile device interaction. During this time, the network should be in preparation for the transition to a secondary state. The purpose of the secondary state is to preserve the communication between devices where possible and to preserve the internal state of all devices that make up the smart space in the event of its destruction. The secondary goal should also support the ad-hoc association of first responders who might wish to help mitigate the situation through human intervention, or should facilitate the identification of the location of people in relation to the system and where possible, allow communication of their state.

1.3 Research Aims and Objectives

This thesis aims to design routing and data distribution protocols to support the Machine-to-Machine interaction of resource-constrained devices that will form the future Internet of Things. Such protocols must support the reliable communication of machine state between devices whilst the system is failing through state redundancy, as well as the communication between devices and the Internet during normal operation. The solution must consider the resource-constrained nature of the devices that operate in this space, but of greater concern are the

mechanisms to support the preservation of network state preceding, during and following failure.

The main overarching aim for this this thesis is to improve the reachability of the devices that will make up the Internet of Things. The approach should provide device reachability during mobility and accommodate the sleep state that devices of this type experience. The proposed scheme should ensure the reachability of a devices state, even if the device has been lost or become detached from the wider internet and mitigate the impact of topology structure during initialisation and device failure.

The specific objectives of this thesis are as follows:

1. To design an architecture to support the global machine-to-machine communication between low-power wireless devices. The mechanism of communication must support the global reachability of devices, as well as their mobility, as they transit between wireless domains. It must provide a mechanism to support state preservation of individual wireless devices whose current wireless domain is disconnected from the Internet and experiencing device failure.

2. To design a GHT compatible and storage mechanism to support state preservation within low-power wireless networks. This protocol must be able to manage local failure within the wireless sub domain and support mechanisms for independent state discovery.

3. To design an algorithm to resolve the Address Space Balancing problem associated with identity provision in wireless Geographic Hash Tables. The algorithm should provide an identity to a device that is relative to its position within the environment in order to preserve the geo-separation of data. The algorithm should minimise the imbalance of the key space allocation and provide good reachability characteristics.

4. To design an autonomic distributed algorithm to identify and respond to network failures within low-power wireless networks that are utilising Geographical Hash Tables as a distributed storage and communication platform. The approach should consider the low-power nature of devices by addressing routing and storage imbalance that can occur near the border of the failure.

The main aim of this thesis is to improve the reachability of individual devices or device state if the physical device has been damaged or is in a sleep state. Objective 1 relates to the development of an architecture to support the mobility of devices and support the application of GHT within the wireless domain. Reachability will be improved through the use of indirection. Objective 2 outlines a mechanism that can be used to store device state in a redundant way within a GHT, through geospatial state distribution. This mechanism can be used independently or as a component of the implementation of the architecture detailed in objective 1. The address allocation mechanism relating to outcome 3 can be used by the implementation of the architecture and also apply the keying mechanism in 2. The address allocation mechanism can be also be used independently. Outcome 3 will improve reachability through ensuring optimal

balancing of device state in the GHT. The autonomic distributed algorithm outlined in objective 4 can be used to mitigate the failure of a topology initialised in outcome 3, but can also be used with other greedy forwarding routing protocols. Rebalancing the topology and improving reachability following failure will ensure that a greater proportion of nodes state are reachable and that nodes surrounding the failure do not become overburdened.

1.4 Contributions to Knowledge

This thesis presents the following novel contributions.

- **Internet Indirection Architecture for the Internet of Things**:
  This thesis details a novel approach to routing within the Internet of Things, specifically proposing the use of global indirection to support the mobility and accessibility of Internet of Things devices whilst they are asleep or mobile. This contribution is provided by the architecture description of Objective 1.

- **Rendezvous Redundancy Communication Protocol for Geographic Hash Tables**:
  This thesis details a novel approach to the distribution of rendezvous state within a wireless networks that deploy Geographic Hash Tables. This enables individual devices to conduct rendezvous communication whilst simultaneously providing a level of redundancy through the placement of data that is geographically separated in the topology. This contribution is used to generate identities within the keying mechanism required for Outcome 2.

- **Anchor Free Localisation of GHT, Addressing Reachability and Balancing Concerns using Genetic Algorithms**:
  This thesis provides a novel approach to the problem of balanced identity assignment in Geographic Routing when dealing with irregular topologies. This novel approach applies genetic algorithms to a spring force layout utilising alternating fitness functions to provide an optimal topology.

- **Correcting GHT Imbalance through Topological Isomorphism:**
  This thesis details a novel algorithm to address the issues post-failure in GHT where the border of a fault becomes a routing and storage bottleneck. The topological isomorphic function distributes both routing and storage load away from the fault border.

The novel contributions made in this thesis can be used to instantiate an instance of the architecture that is the focus of objective 1 or they can be used independently as part of existing systems.

1.5 Publications

To assist in the development of this thesis a number of publications and a journal article have been presented:

Attwood, A. Lamb, D. Abuelmaatti, O. Position-relative identities in the Internet of Things; an evolutionary GHT approach. IEEE Internet of Things Journal. Accepted subject to revision.

Attwood, Andrew, Madjid Merabti, and Omar Abuelmaatti. "IoMANETs: Mobility architecture for wireless M2M networks." GLOBECOM Workshops (GC Wkshps), 2011 IEEE. IEEE, 2011.

Attwood, Andrew, Omar Abuelmatti, Paul Fergus. M2M Rendezvous Redundancy for the Internet of Things. In Developments in E-systems Engineering (DeSE), 2013.

Attwood, Andrew, Madjid Merabti, Paul Fergus, and Omar Abuelmaatti. "Sccir: Smart cities critical infrastructure response framework." In Developments in E-systems Engineering (DeSE), 2011, pp. 460-464. IEEE, 2011.

Robert Hegarty, David Lamb, Andrew Attwood. Digital Evidence Challenges in the Internet of Things. Tenth International Network Conference (INC 2014). Plymouth, UK, July 8-10, 2014. ISBN: 978-1-84102-373-1

1.6 Thesis Structure

This thesis commences with the provision of a background review of related work in the areas of wireless communication, offering a detailed survey of sensor networking routing protocols. This thesis reviews the use of IP in low-power wireless systems and further provides a detailed study relating to mobility support in wireless domains. Following the background, additional context for this work is provided, detailing an approach to the management of Future Cities as well as providing insight into the scenarios that would be faced by future cities' architectures and the wireless sensing systems that will provide sensor actuator capability, enabling The Internet of Things. This thesis then presents an architecture to support the Internet of Things. The following use cases are considered: Wireless Devices attached to their home location, Devices that are mobile in transit between non-home networks, and finally where a network is failing and needs to operate in a fully distributed model to best preserve the continued operation of the system. This is followed by the specification of an alternative approach to providing state

preservation for wireless devices in Geographical DHT using distributed rendezvous keying. Following the definition of the rendezvous communication mechanism, a solution for balancing Geographical DHT spaces is provided. The rendezvous method is used to generate data to evaluate the approach. A distributed method to resolve faults in wireless Geographic DHT is then proposed, followed by an evaluation of the approach through simulation. To finalise the thesis, a summary of the work is provided, identifying its contribution. To conclude, future work is detailed.

In the remainder of this Thesis:

- Chapter 2: Background

  This chapter provides the reader with a background survey of related work and technology. The current state of the art relating to wireless routing protocols is reviewed, with consideration to Distributed P2P communication and wireless sensor networks, as well as an introduction to the Internet of Things.

- Chapter 3: Literature Review

  This Chapter will provide a review of the Literature that the work in this thesis extends. Specific issues relating to existing approaches will be identified as we look to accommodate the requirements of critical infrastructure and the Internet of Things.

- Chapter 4: Mobility, Routing and State Redundancy for the future Internet of Things

  Chapter 4 details an Architecture for the Internet of Things that utilises rendezvous to provide support to mobile low-power devices. An evaluation that shows that the wireless requirement of the scheme can be met by low-power devices is provided.

- Chapter 5: Rendezvous Redundancy for the Internet of Things

  Chapter 5 details a data distribution and rendezvous mechanism for wireless Sensor Networks. A comparison between the proposed rendezvous mechanism and random distribution methods characteristic of overlay DHT is provided.

- Chapter 6: Balanced Wireless Localisation using Evolutionary Algorithms

  In this section, the concept of Localisation in Wireless Mesh Networks is introduced to provide identities in situations where other locations services are unfeasible. The use of Genetic Algorithms to improve graph-based Localisation schemes when provisioning identity spaces for GHT storage schemes is proposed.

- Chapter 7: Correcting GHT Imbalance

  This Chapter details the design and evaluation of a Topological Isomorphic function for balancing load in Geographic Hash Tables (GHT). The chapter starts by detailing how a Two-Dimensional Identity Space is transformed following system failure to reduce routing and data storage imbalance. Finally, a design and evaluation for a routing protocol that utilises the Topological Isomorphic function is proposed.

- Chapter 8: Conclusions and Future Work

  This work is concluded by providing the findings and a summary of the supporting results. Contributions provided by this work will be reviewed along with a proposal for future work.

Chapter 2

Background

Providing reliable machine-to-machine communication between low-power Internet of Things (IoT) devices encompasses a diverse range of technologies. Wireless endpoint devices are usually a combination of low-power embedded processors and low-power radios that usually exhibit limited range and bandwidth. A primary concern of IoT devices is the management of those finite resources to maximise the available battery capacity. Often, devices are expected to operate for a number of years without the expectation of user intervention. This is, of course, down to the individual application characteristics or the device goal, and it could be required that a device forgoes longevity to support a system during times of emergency.

Devices can be powered from a range of renewable sources, such as solar or kinetic energy harvesting: for example, devices might be installed in a field to report on environmental conditions. Devices deployed in such a way might be expected to operate for a number of years when taking samples every 30 minutes, transmitting the information back to a base station every day. It would be common to see such a device being powered by solar energy. Alternatively, a device on a door mechanism could be powered by the kinetic energy produced when the door is opened.

Often, wireless sensing devices are installed in collections referred to as Sensor Networks. Usually, these low-power devices cooperate so as to enable connectivity to a base station that, in turn, provides connectivity to either a private network or the wider publicly accessible Internet. Sensor Networking encompasses the design of distributed routing protocols that provide identities (if required) to devices and a mechanism to transmit information between devices or between devices and the base station. This is referred to as wireless multi-hop communication. The type or routing protocol uses a particular sensor network where deployment is governed by the requirement or goal of the sensor network. The communication requirements can vary between implementations, with some networks requiring source to sink communication whilst others require node-to-node communication. There are also schemes that utilise a rendezvous communication method that treats the sensor network as a distributed storage space.

Traditional sensor networks can be seen as separate islands of devices connected to external systems by the means of a proprietary gateway. In this instance, the device usually has a wireless interface to the sensor network and a wired interface to a computer, wired infrastructure network or WAN terminal, as shown in Figure 2.1. The Internet of Things is an evolution to the sensor networking approach, where devices are provided with globally reachable Internet Protocol addresses to wireless sensing nodes, and instead of using a proprietary, gateway devices are interconnected to the wireless domain using a router. This provides a mechanism for end-to-end device communication, utilising standards based technologies, such as HTTP over IPv6.

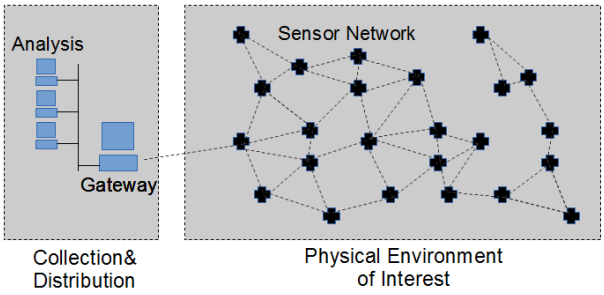


Figure 2.1: Sensor Network

A proportion of the devices attached to the Internet of Things are expected to be mobile; that is, a device with an established identity will move from their home network to associate with a second wireless domain. Device mobility in the IoT will be commonplace, as has been seen with the application of IoT technology to Body Area Networks and Vehicular Networks, with both use cases having mobility requirements. Static devices will also experience similar characteristics to mobility if their home network fails and they associate with other wireless domains to maintain connectivity. IoT devices should exhibit autonomic characteristics in their objective to maintain a machine-to-machine (M2M) connection. This should result in IoT devices making every effort to communicate their internal state and, in some case, to preserve this state with others if mobility and/or possible device destruction would severely hinder the capabilities of the device to do so at a later point in time. To ensure the availability of a device's state in situations of failure—either of itself or a percentage of its environment—the rendezvous state can be distributed using a data redundancy function.

The IoT will provide the mechanism to extract information from the environment, but it does not address where or how the information will be managed. It is possible for IoT devices to be installed within a location to perform a specific localised task, for example provide information for a heating system. However we expect IoT devices to interact and provide information to higher level systems to provide increased global knowledge of the environments that we interact. This work is referred to as Future Cities, the manifestation of regional aggregation of data that can then be accessed at national and international level. The scenarios that are typical of sensor network deployments usually result in the interaction of many services such as Fire, Ambulance and local government. It is envisaged that Future Cities platforms will provide a standards based mechanism to enable the exchange of information. These platforms will also be responsible for the continued operating of those components of the city that are deemed critical. Critical

infrastructure is defined as assets that if removed will affect the populous in an adverse way. For example a chemical factory may produce materials that are critical to cleaning water supplies. The factory may also cause a hazard to the local populous if there was a fire. Future Cities should manage the response to issues arising at facilities to best enable the continued operation and mitigate the effect on those who leave nearby.

This chapter provides a background review into the latest relevant research relating to sensor network and Internet of Things technology, as well as future cities developments.

2.1 Wireless Mesh Routing

Mesh standards identify the ways in which a node will interact with the physical interface so as to enable the successful propagation of a frame from source *S* through intermediate nodes *I* to be delivered at destination node *D*. In an effort to accomplish this, all nodes will need to make best use of the interface available. In a wireless environment, this a major limiting challenge as the interface is usually lossy due to external noise and internal channel issues, such as frame collision. This section will review the two main wireless standards.

2.1.1 Wireless Mesh Networking - 802.11s

In a traditional deployment, 802.11 networks are commonly used to connect a number of mobile nodes back to a wired infrastructure. The most common use is the provision of wireless Internet access to a number of mobile devices. This application is widespread, and can be seen in homes and offices around the world. Largely traffic originates externally from the network and is broadcast from the access point to a wireless node. This is commonly referred to as an asynchronous traffic flow. Usually, there is little node-to-node communication; however, this is largely based on implementation.

The coverage area of individual access points is limited and, as the distance between access point and mobile device increases, the data rate will decrease; this is owing to the adaptive nature of the modulation schemes implemented in the physical layer [1]. Providing Internet coverage for a large area, e.g. town centre, has proved challenging as infrastructure needs to be installed at the site of each access point. Therefore, it would be better to avoid the installation cost of the infrastructure and instead use a wireless back-haul. In an effort to meet this need, the IEEE are working on the 802.11s draft standard for wireless Mesh networking [2]. This amendment to the 802.11 standard recognises the need to construct an ESS (Extended Service Set) without utilising a wired DS (Distribution System) [3].

The following are the key components of an 802.11s network:

- Topology Formation: How nodes join and leave the network
- Routing in MAC: How frames propagate the network using Layer 2 Addressing over Multiple Hops
- MAC: Multi-Channel radio aware propagation of frames and congestion control.
- Internetworking: Connecting wireless Mesh networks to infrastructure and other wireless Mesh networks.

In order to construct a 802.11s network, a number of new nodes are required. Mesh Points interconnect with Mesh Access points to form a wireless DS, which enables MSDUs (MAC Service data units) to transit between Wireless Mesh Access Points, who, themselves, can also act as Mesh points within the wireless DS. To connect to a non-802.11s network, there is the requirement of a portal that is combined with a Mesh point to form a (MPP) Mesh Portal Point. The SSID found in 802.11 networks has been replaced with the Mesh ID in 802.11s. The node seeking to join a Mesh must supply a Mesh profile that matches the existing profile in use by the Mesh. It is imperative that node joining has the same security and path selection protocol identifier [2]. When a Mesh point activates, it conducts active or passive scanning if no Mesh is found, meaning the MP will then create a new Mesh network selecting a channel precedence value based on the boot time of the MP plus a random number. If this new Mesh detects another, then the channel with the highest precedence value is chosen. The draft standard defines a Mesh network as two or more nodes that are interconnected via 802.11 based wireless service to comprise a Wireless Distribution System (WDS).

To support a topology where multiple devices support multiple channels, a Unified Channel Graph approach is adopted. A unified channel graph is a set of nodes interconnected on the same channel within the same network. However, its implementation in 802.11s is lacking where fast-channel switching is required [2].

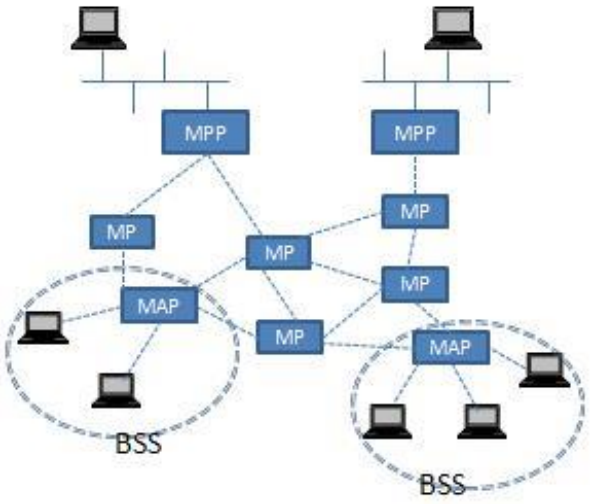


Figure 2.2: 802.11s Mesh Network

In order to establish and maintain a network, Mesh Action frames are transmitted, which support the formation and maintenance of the relationship between all nodes. The transmission of frames between Source and Destination nodes is accomplished within the Mesh boundary using Mesh under protocols within the mac layer. The current draft specification states the mandatory use of HWMP (Hybrid Wireless Mesh Routing Protocol). This routing protocol is hybrid in that it utilises both on-demand and proactive components to build routing knowledge.

On-demand routing is better suited to the highly mobile section of the network, e.g. mobile broadband users in a city centre. Attempting to build and keep routing tables up-to-date for a large number of mobile devices would consume much of the bandwidth and processing capabilities of the network and, with the addition of nodes and increase in their mobility, would degrade network performance even further over time. Instead, routes for such mobile nodes are discovered as required. The discovery of these routes in 802.11s is achieved with the application of AODV with enhancements [4].

Where a Mesh network shows stability, it is better to use a proactive routing protocol. It is usual to find that the stable areas of an 802.11s would be access points that would form the backbone of the Mesh topology. Any node that shows a defined level of stability could form part of the proactive routing section of the Mesh.

In order for the routing protocol to determine the best link, there is the application of the airtime metric. The airtime link is composed of the bit error and modulation rate of the link. Each MP can provide support for synchronisation to reduce the effect of collisions. In order to maintain synchronisation, a global view of timing is required. Each MP achieves this using time-stamp and offset information contained within beacon frames. Power-saving in the Mesh can be achieved through each MP going into a low-power or sleep mode. Sleeping devices have a detrimental effect on the topology of the network; this can be augmented through synchronisation. If nodes can agree on common times to wake for routing tasks, this would minimise network disruption

Access to the MAC layer uses the EDCA scheme specified in 802.11s, which provides differing priorities to different services. Additionally, a point coordination function (referred to as MDA (Mesh Deterministic Access)) can be utilised between neighbour Mesh Points to negotiate the allocation of a number of MDA Opportunity slots. Each slot is 32 μ and, once negotiated, the transmitter is referred to as the owner of the slot group. Notably, a non-owner will have to set their NAV (Network Allocation Vector) to the end of the restricted period [5] .

Issues relating to the 802.11s standard:

- There is a lack of fairness or opportunity for nodes that are distant from the network core to have the same quality of service available to those closer to the network core.
- Mobility of MAP would result in route staleness in proactive routing tables.
- Lack of route prefixing could result in large routing tables.
- Any increase in station-to-station communication could result in a large reactive routing overhead.
- Reliance on tree route nodes gives topology a single point of failure.

2.1.2 Low Power Wireless Mesh Networking - 802.15.4

The IEEE 802.15.4 standard details the physical land MAC layer for Low PAN (low data rate low-power personal area networks). This standard has been produced to fill a growing requirement for low-data, low-power communication networks. The standard was not designed specifically with sensor networks; however, its characteristics make it an ideal choice for such an application.

Commonly, 802.15.4 is seen as the lower levels of the Zigbee stack; however, there are a number of alternative MAC and network layer solutions.

In an effort to support the Internet of Things [6], it is required that 802.15.4 supports a range of devices that are not commonly found attached to networks in the past. To achieve this, 802.15.4 supports two device types.

- Fully Function Devices:
    - Can be personal area network coordinator, needing at least one in every topology type.
    - Perform coordinator function required for each branch in the tree topology.
    - Client functionality.
- Reduced functional device:
    - Has the same properties as the FFD client but does not have capabilities to perform advanced functions. It is envisaged that these devices have very small resources (could be embedded in light switches, food cartons, etc.).

802.15.4 is capable of forming two topology types, although one of those can be extended to increase the topology count to three notional topologies. The first topology—and notably the most simple—is the star, which has a single PAN coordinator with a number of RFD nodes. In this topology, all device communication is conducted via the central PAN coordinator. One implication of this is that the FFD must have adequate power resource to service each node. Different approaches have been taken to reduce the burden on the elected FFD, such as LEACH and TEEN [7][8].

This topology has a single point of failure; however, it is simple and would have low overhead with no need for higher layer protocols. The star topology can be extended to form a cluster tree. There is still only one PAN coordinator; however, each branch must be formed using a cluster head (FFD coordinator).

The second supported topology, the cluster Mesh, has a single PAN coordinator and number of FFD and RFD nodes. There is no prescribed connectivity pattern, as seen in the tree topology. In this case, a node can communicate with any adjacent node but, without the use of a higher layer protocol multi-hop transmission, this is not possible; in actuality, either route over or Mesh

under routing would need to be implemented on top of the 802.15.4 stack in order to support cross Mesh frame propagation.

The common data rate for 802.15.4 is 250kbps using the 2.4 ghz band. This frequency range is usually crowded so the PHY layer uses DSSS (Direct Sequence Spread Spectrum) Spreading technique to counter the effects of channel overuse.

Issues relating to 802.15.4:

- Frame loss results in retransmission from the source in large networks with lossy links, which would result in a large retransmission overhead.
- Small frame size will impact on the ability to carry additional IPv6 services and tunnelling information that mobile IPv6 and NEMO rely on.

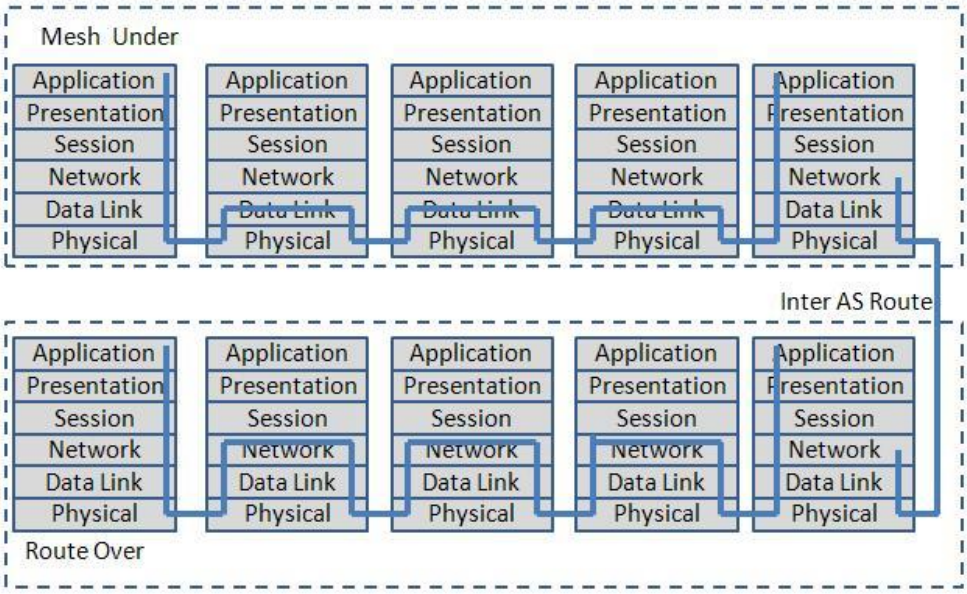


Figure 2.3: Route under Route Over 7-Layer Model

Mesh routing protocols are required to formulate a path for data propagation between node *S* and node *D*. This can be achieved either by a Layer 3 Protocol (Route Over) or by a Layer 2 Protocol (Mesh Under) [9] . Routing protocols can be further categorised as either proactive or reactive [1], where proactive routing is best applied to networks with reduced mobility whereas reactive protocols work better in a highly mobile environment. However, in some implementations, it is sometimes better to apply both techniques to a single protocol where there are networks with areas of stability and zones of instability.

Mesh underutilises the underlying MAC Addresses to form routing tables in the MAC layer. Route over requires the addition of an IP stack on the device. IPv4 limited address space is seen as insufficient to cope with future demands, meaning the IETF have worked toward creating the 6lowPAN standard [10]. The 6lowPAN standard defines a number of techniques, such as header compression, so as to enable the transmission of IP within the limited 250-byte payload of

802.15.4. The two common operating systems used by the Low PAN community are Contiki [11] and tiny OS [12], both of which support 6lowPAN [13].

As stated earlier 802.15.4 supports the formation of star extended tree and peer-to-peer topology; on the other hand, however, it does not specify any mechanism for end-to-end transmission in Mesh environment over 1-hop distance. Zigbee [14] adds such functionality, providing a full protocol stack to enable quick deployment.

2.1.3 Reactive Routing

Where networks exhibit high mobility, the cost of maintaining routes would overwhelm the available bandwidth. Reactive routing saves us the overhead of maintaining the current topological state of the network at the cost of a delay in transmission when there is data to transmit. In a reactive routing environment, the path is established on demand, although routing protocols usually account for the fact that some nodes may have partial knowledge of the network from a recent reactive route update. This information can then be used to build new routes limiting the broadcast of route request frames.

Dynamic Source Routing protocol [15] supports the following functions:

The formation of a route between node *S* and node *D* where node *S* has no knowledge of a route to *D*. DSR also supports the maintenance by node *S* of a previously discovered route. In the case of route failure, node *S* can use network knowledge to repair the route or otherwise to initiate a new route discovery process. DSR is an on demand protocol requiring no periodic updates for route formation or maintenance. Importantly, nodes can cache multiple routes to a single destination so that, in the case of route failure, alternative routes can be applied without maintenance or rediscovery.

When node *S* has a packet to send to node *D* it checks its route cache to see if it has a previously learnt route between *S* and *D*. If no route exists, node *S* broadcasts a route request to all nodes in the wireless range. The route request includes a unique route ID generated by the source—a route record that will contain a list of the intermediate node addresses and *S* and *D* addresses. If the node receiving the route request is node *D*, then *D* responds with a route reply that contains the list of intermediate nodes that have accumulated in the received route request. The source then caches the route to the destination node. If a node receives a route request with the same ID and source address or finds that its address is already in the route list, then the packet will be discarded. The node can otherwise add its own address to the intermediate node list in the route request and rebroadcast the frame.

In an effort to avoid the issue of unidirectional links, node *D* will not reverse the route from *S* to *D* to get back to *S*. Instead, node *D* will refer to its own cache to see if it has a route to *S*. If not,

D will initiate its own route request to *S*. Separate routes are discovered as it is not correct to say that transmission between two nodes will be bidirectional due to antenna or signal propagation issues. In order to avoid infinite route discovery between *S* and *D*, route request from *D* is attached to the route reply. When *S* sends a packet to *D*, each node is responsible for confirming delivery to the next node in the list. This could use data link layer capabilities, network layer confirmation or a passive system, such as detecting retransmission [16], which would be possible on bidirectional wireless links. If the intermediate node fails to receive a confirmation of receipt by the next hop, a route error packet is sent back to node *S*. This route will then be removed from the route cache on node *S*. The retransmission of the packet would be controlled by other layers of the protocol stack.

Intermediate nodes can attempt to salvage the lost frame after the route error frame has been transmitted. If the node has an alternate route to node *D,* it can then replace the failed route with its own cached route. If a node detects a frame that is being transmitted between intermediate nodes but it finds that its own address is contained in the list of nodes, it can shorten the route by sending a gratuitous route reply to the source. When route errors are received by the source, they can be attached to subsequent route requests to the destination. This will inform the source neighbours that the route has an error and should be removed from its own cache. This will prevent intermediate nodes from responding with incorrect failed route.

Nodes can add routing information from route request packets that they receive; however, this may cause problems when the route request has propagated a unidirectional link. If an intermediate node receiving a route request has a route to the destination node, it checks that its route has no duplication of intermediate nodes and sends a route reply back to node *S*. So as to avoid a route reply storm from neighbouring intermediate nodes, each intermediate node will back off for a random time interval to see if the node *S* has transmitted a packet to node *D* using another neighbour response.

Issues relating to DSR:

- Overhead when finding a route and double-overhead when piggy-backing to find the reverse route from *D* to *S* would not be suitable for the small frame size of 802.15.4.
- Route maintenance over unidirectional paths will result in multiple route requests to return a route confirmation to previous intermediate nodes. If route requests flood the network in a highly mobile environment, this could impact on the available bandwidth.
- Gratuitous route requests when a node that intercepts a packet with its own address later in the route could have been mobile itself, and is no longer part of the route in transmitting the gratuitous route response, giving incorrect topology knowledge to the source.

Adaptive on-demand distance vector routing protocol is reactive protocol forming routes to hosts as required [4]. The designers of AODV note scalability issues relating to maintaining routes in an early protocol named destination-sequenced distance vector (DSDV). DSDV requires periodic broadcasts for the dissemination connectivity information. The overhead of this communication grows as (n^2). AODV uses broadcast route discovery in a similar way to

DSR; however, as opposed to building a source route list in the request packet, AODV uses route table entries on the intermediate nodes. When source node *S* is in demand routing mode and wishes to transmit a frame to destination node *D*, Node *S* transmits a RREQ frame that will be received by Intermediate nodes I. When an I node receives the RREQ frame, it checks the source ID and broadcast ID of the packet against previously received frames. Notably, if there is a match, the frame is discarded. If the frame is not for the receiving node, the packet hop count is incremented and the packet is forwarded. The node will then record the destination, source address, broadcast ID, expiration timer and source sequence number.

As the RREQ propagate the network, the reverse path is formed with each node having the preceding nodes address in the route. The RREQ will reach a node that is the destination or has knowledge of a route to the destination node. If the destination sequence number is less than the stored sequence number of the node, the process of broadcasting should continue. If the route is good or the node is *D,* a RREP should be sent using the established reverse path. Nodes in the route to *S* are able to update the forward routing information to node *D*. Nodes periodically send HELLO messages with the objective to detect node movement. If the loss of a node results in broken route, then a RREP message can be sent upstream to the source's node, which then can issue a new RREQ with an increased sequence number.

Issues relating to AODV:

- The transmission of HELLO messages can consume network bandwidth.
- Resources on the nodes would be consumed to store routing tables as there is no way of prefixing sets of nodes.
- A route cannot heal itself or make improvements.

2.1.4 Proactive Routing

Proactive routing builds routing information about all nodes in a Mesh network before the data is ready to transmit. Network formation usually occurs when the network is switched on. Routing updates are disseminated through the network, and each node creates a table of all the other nodes on the network and has some knowledge of how to reach them. Usually, these routing messages are transmitted periodically or if a node detects a change in its status. In the case of wired systems, this is usually a link failure.

Optimised Link State Routing Protocol (OLSR) is a reactive link state table-based routing protocol. However, unlike a link state routing protocol, OLSR sends only routing updates to a subset of its neighbours [17]. These neighbours are referred to as multi-point relays. Messages are generated periodically and are not based on events and received by all neighbours; however, they are forwarded by only multi-point relays. A node selects its multi-point relays so that it has contact with all of its two-hop neighbours. The smaller the multi-point set, the more optimal the network. Each node periodically broadcasts its multi-point neighbour's node. Nodes use this information to calculate routes, resulting in routes comprising multi-point relays with each link validated to be bi-directional. HELLO messages are transmitted but not relayed to one-hop

neighbours: if a neighbour detects its own address in the frame, the node can then mark that link as bi-directional. The node uses HELLO information to determine the minimum set of one-hop neighbours that can contact all two-hop neighbours. These are the nodes elected to become multi-point relays. In order to build the intra-forwarding database, each node broadcasts a Topology Control (TC) message through it multi-point relays. The TC message contains the list of nodes that has selected the sender as a multi-point relay. All nodes on the network can then use this information to build the network topology.

Routing tables are calculated using sets of connected pairs. If a node seeks to find a route to *D*, it finds a set connected (D,B), then intermediate connections to *S* (B,C) (C,S). Once this route is found, the packet can be sent to the next node (C).

Issues relating to OLSR:

- The election of multi-point relays should consider other properties.
- It may be better to select sub-optimal relays based on power requirements.
- Mobility needs to be related to the frequency of control messages.
- Highly mobile sections could flood the network with TC messages.
- The inclusion of the addresses of all nodes in the path would not scale well and would not be suitable for large 802.15.4 networks owing to limited packet size.

2.1.5 Hybrid Routing

Reactive protocols work well in networks with high mobility. In highly mobile networks, trying to maintain path information between all nodes would be unrealistic and would require a frequency of routing updates. Proactive protocols work well in Mesh topologies with low mobility. Most Mesh networks that will operate in the fringe Internet will have areas of high, moderate, low and no mobility. In such networks, using one scheme would not be effective.

Hybrid wireless Mesh routing protocol, as used in 802.11s, utilises aspects of AODV and OLSR [18]. AODV-reactive routes are established using Path Request (PREQ) messages. More specifically, designated Mesh Points form the proactive section of the network, thus building a proactive tree. The tree is maintained by the designated MP transmitting periodic proactive PREQ messages and root announcements (RANN).

Issue relating to hybrid protocols:

- How can staticity be pushed into nested networks? Mobile Mesh networks may have their own proactive sections; there is no mechanism to merge or take into account another network's proactive or reactive sections.
- Proactive sections may cause traffic imbalances at the edge of the dynamic region, causing battery usage on those presumably mobile devices.
- Semi-proactive protocols could be investigated to provide an awakening node a routing table of its last known state and to give neighbours updates concerning neighbours'

current tables. This could reduce radio and processor usage; on the other hand, however, this could provide stale information relating to neighbours outside of the providing neighbour's radio range.

2.1.6 Other Routing Considerations

Multi-path routing is concerned with providing, either fully or partially, separate paths between source and destination. There are a number of reasons for providing this diversity [19]:

- Path protection in the case of link/node failure
- Load balancing on the node
- Conservation of battery power
- Increased bit-rate.

If using hop count, this will not take into account the problems of energy use. Moreover, single hops at Layer 1 hide the reality of multiple Mesh under hops that occur transparently to Layer 3. Fault-tolerant routing-required back up paths should be established due to lossy nature and mobility of nodes.

Hierarchical routing or clustering is the process of grouping local nodes into a stub of a larger network—usually a tree topology. In an effort to help address the scalability problem, cluster head election schemes have been developed to fairly detect new cluster heads whilst taking into consideration various mobility metrics.

The mobility of nodes will affect the topology of the Mesh, which could cause increased packet loss and additional routing protocol overhead. Other schemes [20] exploit this mobility through the use of multi-user diversity; however, this particular scheme is limited to data that can be buffered and therefore is not considered suitable in the context of real time applications.

Multi-user diversity waiting until the mobility of an object provides a better topology for the transmission of data, but there would be a need to calculate that holding on to the data would not cost more than waiting for transmission. This has important consequences in terms of Mesh inter-networking as, in terms of the cost of transmission, although autonomous nodes is high—meaning that deferring transmission until an optimal autonomous system inter network topology is present may be preferable—this could be based on energy or security considerations. Obviously, real-time urgent data would need to switch to a more proactive scheme and could not rely on the buffering process.

2.2 Internet of Things - Routing over Lossy Links

The IETF are working on a new standard referred to as ROLL (Routing Over Lossy Links) [21]. The routing algorithm being developed is generic, meaning it does not target a single technology, and has a core set of functionality; this will enable customisation of the protocol for different

implementations [21], as defined in the IETF Internet draft. RPL networks can run multiple RPL instances, with each instance having a number of DAG (Directed Acyclic Graphs). A single RPL instance has a specific objective function that ties together metrics, constraints and optimisation objectives. Each Graph has at least 1 DAG root; in the event of a single DAG root, this is referred to as a DODAG (Destination Oriented DAG root). Each node in a DAG is ranked as to its distance from the DODAG. Traffic is bound to an RPL instance by inserting the RPL instance number into the flow label of the IP6 header. Contiki has been tested with RPL and IPv6, and has shown that a sensor network can operate for several years, utilising the skymote with standard 3000 mAh AA batteries [11].

One of the major concerns surrounding the application of IP and IP routing on wireless Mesh networks relates to the limited memory resources on the target devices: for example, one popular device the TelosB sensor has 10kb of RAM, 48kb of program memory and 1mb of flash memory used for data logging etc, as can be seen in Table 2.1, where the Contiki operating system has IPv6 and RPL needs 3224 bytes of ROM and only 800 bytes of RAM.

Issues relating to ROLL:

- There seems to be little consideration to NEMO and the concept of network mobility.
- It utilises a distributed approach to generating routing knowledge.
- Code density can be maintained once additional IPv6 services are added, such as Mobile IP and NEMO, and their associated state.

Module	RAM (bytes)	ROM (bytes)
Generic IPv6 routing	420	484
RPL packet generation and parsing	2	1316
RPL protocol logic	378	1074
RPL timer handling	0	350
ContikiRPL Total	800	3224

Table 2.1: RPL Implementation Size [11]

2.3 Future Cities and the Protection of Critical Infrastructure

The protection of critical infrastructure is regarded as one of the 21st Century's greatest challenges. Critical Infrastructure refers to any entity that provides a service to the maintenance of the wellbeing of the populace and something that, if removed, would cause harm or serious disruption. For example, in the US, one of the greatest critical infrastructure problems is the supply of electricity. There have been concerns surrounding the capability of the US electrical grid for a number of years, which, in 2003, culminated in a series of failures in the north east, causing serious disruption. Telecommunications is also seen as a key critical infrastructure component. In 2004, a fire in a tunnel under Manchester City Centre in the UK knocked out communications for a number of banks and businesses in the area.

Systems upon which individuals and organisations rely are increasingly being interconnected, and the interdependencies between systems, however small, can result in a service being deemed as critical. Failure in even the smallest component of a critical infrastructure can lead to a chain reaction that causes widespread disruption. Systems in close geographic proximity to a critical system but which are not deemed critical themselves can have an impact when failing on critical infrastructure. In April 2011, for example, a fire in a nearby scrap yard caused the shutdown of the M1 in the UK for 6 days. This resulted in major delays and had a significant financial impact.

Responding to an event and minimising its effect on critical infrastructure is essential; this involves providing real-time information concerning the state of the system and the effect its failure will have on surrounding systems. The responding system will require sufficient information to assist in the decision-making process. The implementation of Future Cities and Internet of Things technology will result in an increasing number of sensors that can be used to improve the fidelity of the decision-making process when trying to minimise the impact of failure.

In critical infrastructures, control systems are connected via Programmable Logic Controllers (PLC) to Supervisory Control and Data Acquisition systems (SCADA).Using input from a sensor, SCADA can react and invoke a change within the environment to which it has responsibility. Alternatively, operators can be notified of a problem to enable human intervention to alleviate or minimise the impact of failure. System modifications are propagated to actuators via the programmable logic controller (PLC). Connectivity between individual PLC and between PLC and SCADA has, historically, been proprietary; however, the latest generation of SCADA provides TCP/IP connectivity between PLC, SCADA and higher level management systems. It is usual for SCADA systems to only provide real-time state information internally, and to export little information to external systems.

The 2011 earthquakes in Japan have shown the failings of SCADA and the PLC-driven model when a system experiences widespread system failure. SCADA systems and PLC usually provide a controlled shutdown sequence in the event of failure, which is recognised as being in the best interest of the entire critical infrastructure. Communication between sensors in the event of localised or widespread failure could prevent incorrect states being selected and the continued operation of a system until interconnections between systems are restored. Providing connectivity to sensor and actuators to first responders could also provide useful information and thus provide a mechanism to invoke change within the failing system.

The application of Sensor Actuator Networks (SAN) in critical infrastructure is increasing. It is essential that the availability of sensors is maximised so as to maintain confidence levels in the critical infrastructure state. SAN provides an alternative mechanism to extract the state of a failing system. This, coupled with information from the Smart City level, would provide valuable response support data.

First responders to a failing system, such as fire services, ambulances and police, for example, usually have detailed plans relating to critical infrastructure and have visited the premises to ensure they are aware of the hazards and the impact failure would have on surrounding systems.

Looking to the future Internet of Things and the application of smart cities, the quantity of information that could be used to help assess the impact of a failing system in real-time will increase. In order to process this data, computational capacity and the linking of data will be essential, in addition to the automated reasoning and real-time decision-making processes. The combination of semantic sensor web and cloud computing will provide the information and computation required for the detailed analysis of the semantic representation of the system. However, this will rely on the availability of the current failing SAN state to be able to give a full view of the current system's state.

2.3.1 Automated Emergency response

In an emergency situation, decisions are made based on the experience of the responders. This experience is acquired through training, simulation and, most importantly, involvement in real emergency situations. Experience, coupled with the information that the responder has at the time, will assist in the provision of response. Potential sources of information include building plans, maps, and observations and reports from individuals involved in the incident. Simulation results can also be used to identify the probable next state of a filing system. In [22], the authors introduce the FireGrid framework, during which they describe the importance of ahead-of-time and real-time simulation to help assist first responders in the decision-making process. They detail the use of high-performance grid resources to perform computation in parallel in order to support the real-time decision-making process.

Naja et al. [23] provide precautionary notes on the application of automated response and the compliance issues relating to automated response systems. In [24], Shafiq et al. identify the sharing of information between departments as an essential requirement when providing an effective emergency response. To enable this, it is essential that systems are interoperable. Furthermore, in [24], the authors propose an emergency response framework that includes the following components: ontology library, reasoning engine and a workflow interface for visualisation. The framework does not make reference to the use of Internet of Things information sources or how a response system could interface to SAN so as to minimise the impact of failure.

2.3.2 Smart Cities and the Internet of Things

In [25], the authors detail the increasing pressures being placed upon cities through increases in population. Currently, 50% of the world's population live in cities; however, this is set to increase to over 70% by 2020. Smart cities can be viewed as a system of systems; these are interconnected systems providing a function that depends on the pooled resources of its constituent systems.

Within cities, essential services, such as transportation, energy supply and healthcare, exist alongside commercial non-essential services. Manufacturing facilities can also provide components that can be considered fundamental to the operation of critical infrastructure. As cities and organisations look to reduce costs and consolidate their activities, they become tightly

integrated, increasing their reliance on automated machine-to-machine interaction [26], thus creating ever more complex systems of systems.

Smart cities rely on the flow of information to assist higher level decision-making systems [27] Increasing the sensing capability of a city will increase the fidelity of information to the higher level decision-making systems. Increasing the quality and quantity of data will result in an effective and timely response to infrastructure failure, realising that the Internet of Things will provide the levels of information smart cities require.

The Internet of Things involves pushing internet connectivity and services to the devices with which individuals interact [28]. This technology is mainly deployed to Wireless ad-hoc sensor networks that perform cooperative Mesh routing. There has been a great deal of research concerning the development of Wireless Mesh sensor networks. It was considered that the application of IP on constrained devices—which often run for years on a set of batteries and have limited processing, storage and bandwidth capabilities—was unrealistic; however, recent developments have seen the development 6LowPAN [29], a low-power variant of IPv6 that utilises header compression to enable the transmission of IPv6 within the limited 802.15.4 link layer frame. Markedly, 6LowPAN provides an opportunity for every device in the Internet of Things to have a unique identifier. Coupling this identifier with the UDP COAP [30] service that is being developed by the CoRE working group provides a full service URI e.g. coap:// fe80::202:b38e:ac13/pressure. Routing 6LowPAN packets between nodes is accomplished using Ipv6 routing protocol for low-power lossy links (RPL).

Sensor and actuators networks (SAN) are increasingly being connected to the Internet as part of the Internet of Things. SAN need to be linked using System of System techniques relying on trust and federation to create smart cities [31]. The devices need to advertise and discover the environment in which they operate. Devices would be deployed into a Smart City with some pre-planned application, e.g. a light switch controlling a light bulb; however, as a Smart City reacts to an emergency, a device may need to operate outside of its deployed remit. For example, in an emergency, a system might seek to switch on all light bulbs to give better visibility; however, there may be other cross-cutting concerns, such as if the city, at the time of the event, has limited power capacity, thus meaning it might direct the light system to deactivate. It is essential that a response has the appropriate understanding of the current cross-cutting concerns of the Smart City, where devices may be implemented for a single purpose but may, over time, become involved with other cross-cutting concerns, such as energy (smart grid), governance, health and entertainment.

Increasing the number of sensors within a Smart City will increase the quantity of data needing to be managed and processed. The vision of a fully Internet of Things-enabled Smart City is one that enriches machine-to-machine interactions to a point that it requires minimal human intervention, resulting in a city that can adapt and reconfigure itself to deal with failures and maintain a state that is optimised to meet the needs of the inhabitants. In an effort to achieve this, data needs to be represented in a way that enables it to be processed by machines and for machines to understand the relationships between data. The Semantic Web is an approach to providing meaning to the data provided via the Internet, so that machines are able to reason

using the relationships that exist, or can be reasoned to exist, between data. The relationships that can exist between data on the Internet are referred to as linked data. In [32], the authors detail the annotation of sensor data and the application Semantic Web Rule Language (SWRL) to automate the discovery and annotation of geospatial data. The application of geospatial data is essential when coordinating a response based on SAN state information as is the description of the cause and anticipated effect of a sensor's current state. For example, when a gas sensor detects a leak, the response might be to evacuate to a predefined radius; however, the destruction may have an impact on a nearby system. This will, in turn, impact on a critical system. The response should be adapted with the objective to best minimise the impact of the explosion.

Cloud computing provides a flexible approach to the management of information systems. Its adoption has been increasing as both individuals and companies alike look to reduce the total cost of ownership of their in-house computing capacity. Cloud computing enables users to expand their computing capacity to meet the demand; once the demand has passed, this capacity is returned, resulting in expenditure better matching income with a pay-as-you-use service model, thus enabling smart cities to have similar cost relationships and requiring only peek processing at certain times. Future Smart Cities will have varying processing requirements, depending on situations with the city and the quantity of data being provided by Internet of Things devices.

Infrastructure as a Service (IaaS) is a cloud component that provides flexible computing capacity, including processing capability, storage and network capability. IaaS has the potential to provide a Future City platform with distributed access to information so as to enable real-time processing to fulfil the needs of the situation. Smart cities will have unexpected events that will require on-demand infrastructure: for example, flooding events that happen on average every 10 years will require additional computational capacity. Preferably, this capacity would not drain local power resources or rely on the availability of the city's resources. Utilising IaaS capacity on-demand will ensure value for money and will also provide the fault-tolerance required, with systems deployed in geographically disperse locations.

2.4 Summary

This chapter has provided background and identified the challenges associated with routing in low-power wireless networks. This chapter has also highlighted the issues relating to future cities management of devices that will form the future Internet of Things.

The research has found there is little work relating specifically to the reachability of mobile devices and device collections, such as Body Area Networks within the Internet of Things. When considering the low-power and critical infrastructure requirements of this project, the impact of current schemes, designed for the current Internet, typically introduces both single point of failure issues and whilst also adding significant overhead when devices become mobile. Current schemes also require established authentication between routers and access points of different administrative domains.

Protecting critical infrastructures is an important aspect of ensuring the continued operation of Future Cities. It is required that Future Cities be able to communicate with individual Internet of Things devices. It is also required that, in the case of emergencies, devices are still capable of communicating their state with each other and first responders. This leads us to the requirements of an architecture to support Internet of Things devices when they are in contact with a Future City platform but are equally capable of operating in isolation so as to provide a mechanism for communicating between devices (M2M) and to preserve state for post-failure analysis.

The next chapter provides a literature review of the existing research that is related to the main focus of this work.

Chapter 3

Literature Review

This chapter provides a critical review of the literature related to; device and network mobility, indirection architectures, DHT routing and data storage in wireless sensor networks, and address allocation for position relative topologies. This literature review will help identify potential solutions to fulfil the requirements of the research objectives outlined in section 1.3 as well as identify the problems that will need to be addressed in order to meet the objectives of this work. This review will firstly examine the current schemes that support the mobility of individual and device collections with respect to maintaining internet connectivity. This section will then examine work relating to indirection architectures, these schemes look to use P2P overlay technology to support improved reachability through the use of overlay routing schemes. Finally this review will then look at mechanisms to support devices with the wireless domain. Few existing schemes for global reachability investigate the provision of identity to low power wireless mesh nodes. To facilitate identity provision and provide a mechanism for local communication and the storage of information overlay DHT storage and routing schemes for wireless mesh networks are reviewed. This thesis also reviews the specific problem of identity provision in Geographical DHT spaces where the reliance of GPS or other hardware localisation schemes would be insufficient.

3.1 Network Mobility

3.1.1 Layer 2 and 3 Network Mobility

Node and network mobility is a core aspect to this research and relates to the first objective detailed in section 1.3. Node mobility enables a single device to leave its current point of attachment and to join a second point of attachment. This new network location could be a part of the same administrative domain or another domain with which the mobile device home

network has an established relationship. Any current information flows should be maintained during this process.

The IETF propose mobile IPv6 [33] as a solution to single-node mobility problem. This extension to IPv6 provides a mechanism for a mobile device to leave its home network and join a foreign network.When a mobile node enters the foreign network, it is issued with a care of address (COA) by a router in the foreign network. Once the mobile agent has a COA, the mobile agent registers this address with its home agent router. Data from the correspondent is routed to the home agent router, and then tunnelled via the foreign agent router to the mobile agent using the COA. In order to reduce the increased latency caused by the tunnelling of data from the home agent to the foreign agent, the home agent router can give the correspondent node the new COA of the mobile node. The tunnelling process can then be eliminated and the data flow permitted to travel through an optimised route via the foreign router, reducing the overhead on the home agent. This avoids the problems incurred through triangular routing [33].

Mobile IPv6 requires a home agent at the original point of attachment to initiate the connection and provide the mechanism to remove the triangular routing problem. This creates a single point of failure at the home agent, causing an incompatibility with the critical infrastructure requirements of this project. In an effort to avoid the use of encapsulation and thus prevent the triangle routing issue, the correspondent must use a COA provided by the foreign agent. This could cause firewalls between the mobile node and correspondent to block traffic. Authentication would need to be prearranged between foreign and home agents. This would cause additional administrative overhead when establishing the system and could cause further problems for a system that would need to dynamically respond in the face of system failure.

The main issues relating to the use of Mobile IPv6 are:

- The reliance on a home agent.
- Single point of failure.
- Packet overhead associated with authentication.
- Complex procedure to reduce the effect of triangle routing.
- No support for mobile networks or mesh connections.
- Provides a mechanism for node to node communication. Both devices need to be active for the exchange of information.

The type of device mobility can be categorised as either micro- or macro-mobility [34]. If a node moves within access points of the same domain or area of administrative control, this is referred to as micro-mobility. Alternatively, if a node leaves its administrative area to join a separate domain, this is referred to as macro-mobility. Mobility can be managed in two ways, network- or device-centric. Network-centric mobility provides transparent mobility to the mobile device, whereas in the case of mobile-centric schemes, the mobile node initiates and has some control over the establishment of tunnels and the transmission of control messages [35]. Micro-mobility protocols permit mobility within a single network but fail to address mobility between networks, nor do they consider improvements to the path from the correspondent. Macro-mobility considers the end-to-end path but can induce a heavy communication burden to all network

components. Macro-mobility also relies on devices outside of a common administrative area to cooperate. Mobile-centric mobility schemes provide essential data to enable the best mobility routing decision, but can take valuable resources from the devices. In the case of low-power wireless Mesh networks, this would be detrimental to the system. However, network-centric schemes have a greater network view and have the necessary bandwidth and power capabilities to continuously monitor and change state. However, without the mobile device's input, the network-centric scheme would lack important information. Mobile devices must also be able to operate independently of a fixed network point of attachment—something that a wholly network-centric mobility management protocol would fail to achieve.

To support Mobility in the future IoT the following mobility models would be required:

- Macro-mobility - to support devices roaming freely between networks
- Mobile Centric – Nodes must be able to control connectivity and not be reliant on the network to provide connectivity in a controlled way, as this introduces a single point of failure. It would be preferred that a local distributed function to support communication between nodes that a combination of mobile or wireless centric with distributed support. Also that the global mechanism to support mobility be distributed as opposed to being managed by any one endpoint.

Mobile IP sits at layer 3 so can fail to take advantage of the physical topology of the network. A number of protocols have been suggested to reduce signalling latency for route establishment. Fast Handovers for Mobile IPv6 (FMIP) [36] uses Layer 2 triggers to initiate the process of obtaining a new care of address. Past association-FMIP (PA-FMIP) [37] uses past association patterns to replace/augment the triggers provided by Layer 2. This results in a reduction of packet loss by 45% compared with Mobile IP. With the increased use of Multi-Protocol Label Switching (MPLS) to provide Quality of Service (QOS) in IP networks, Mobile MPLS [38] has been developed to reduce signalling overhead in micro-mobility scenarios where MPLS is in use.

Mobile MPLS proposes the use of a label-edge gateway router to create label-switched paths that can take advantages of the high speed of link establishment and QOS available to MPLS technology. Both schemes make use of MPLS as a method of bypassing the routing decisions within the local point of attachment network.

Issues relating to the use of MPLS to reduce overhead of mobility

- Relying on a single technology prevents widespread application. IP is common on the core internet, whereas MPLS is not.
- MPLS limits the improvements to the area controlled by the MPLS administrative domain, often networks do not provide external access to the MPLS domain, limiting adoption.

It is not realistic to consider the modification of existing core internet technologies. In the core Internet, routing between separate administrative domains is accomplished with Border Gateway

Protocol (BGP) [39]. However, BGP is designed for a non-mobile network where autonomous system domains would remain coherent. In a mobile network it would be likely to see autonomous system domains split into sub-domains. In this eventuality, BGP loop detection would disregard routes to one of the sections as the autonomous system number would appear twice in routing advertisements. BGP has a number of issues relating to the growth of routing tables; this would see its adoption in the fringe Internet as restrictive [40].

Internal BGP uses the internal routing protocol of the autonomous system to deliver packets to other designated fixed BGP gateway routers. In a mobile Mesh environment, all nodes are routers, and it may not by reasonable to assume that a fixed subset of Mesh devices could perform the BGP gateway function; this could be for reasons of battery use or device capability [41]. Routing policy would also need to be dynamic. On the Internet, autonomous networks establish peering relationships, often involving high bandwidth data path arrangements, these relationships are not dynamic and would not suite a critical infrastructure where relationships might need to be negotiated in real-time owing to network failure.

It is important therefore to ensure that proposals do not consider core internet technology that is not in widespread use or is inaccessible by different network operators. With this in mind the work in this thesis will evaluate and extend the work in overlay technology. This approach works on top of existing IP technology, ensuring the greatest level of compatibility with existing systems.

An alternative approach is to use a combination of protocols to cope with micro and macro mobility. HAWAII [42] is a domain-based approach essentially providing micro-mobility within the autonomous system domain of the edge network. HAWAII updates selected routers as to the current connectivity of the mobile device by inserting host-based entries for the mobile node. Packets addressed to the IP of the mobile node would not follow the prefix for the subnet; instead, the packet would be routed by the individual route for the single device. Obviously, this scheme would not scale any higher than the Internet fringe, and has a fall-back mechanism using Mobile IP to cope with macro-mobility. Using the approach taken in HAWAII for macro-mobility would result in a fragmented routing table and an exponential increase in routing table entries.

Issues relating to HAWAII:

- It is not feasible to construct individual routes for every node in the wireless domain.
- Utilises mobile IP, introducing a single point of failure.

This thesis needs to address not only the mobility of individual devices but also the mobility of networks. Network mobility could relate to the mobility of body area networks, or vehicular networks that contain main independent devices that are part of the same network.

In order to provide support for multiple mobile devices or mobile network segments, the IETF proposed NEMO [43]. This extension defines the use of a mobile router that communicates with the home agent so as to provide a tunnel for all mobile devices attached to the mobile

router. Mobile routers introduce a single point of failure to the mobile section, and the added responsibility would quickly deplete the resources on the device. Low-power devices typically use Mesh topologies to create more reliable networks that are more power-aware. Mobile NEMO networks would need to nest to provide connectivity to other NEMO networks not in range of the edge network. Mesh networks give a more reliable topology to link together many networks. Mobile Mesh 6LOWPAN [10] networks operating on the fringe of the Internet may need to route though a number of other equally mobile networks so as to obtain connectivity to the edge Internet. Individual Mesh autonomous network domains may split to form non-contiguous areas; this differs from NEMO as it defines each mobile network as an autonomy as opposed to a branch in a collective tree

.

Issue relating to NEMO:

- Requirement for a mobile router would quickly overwhelm a single low power device.
- A NEMO mesh would not enable a nested mesh to obtain connectivity.

Schemes such as NEMO [43] permit connection through additional software on the new router and the establishment of a tunnel back to the home network. NEMO relies on a dedicated mobile router and does not permit further fragmentation. Using Internet Protocol V6 enables all devices to be mobile with the same address they had when attached to their home network. So as to facilitate the mobile devices to fragment into smaller networks, the reliance on the mobile router must be removed; instead, a fair election scheme that will permit the election of devices should be designed, which will need to act as routers for the fragment. If segments are to permit nesting—that is, to permit other fragments that are not a part of its own administrative domain to route packets through (as shown in Figure 3.1)—multiple fragment Edge Routers will be required.

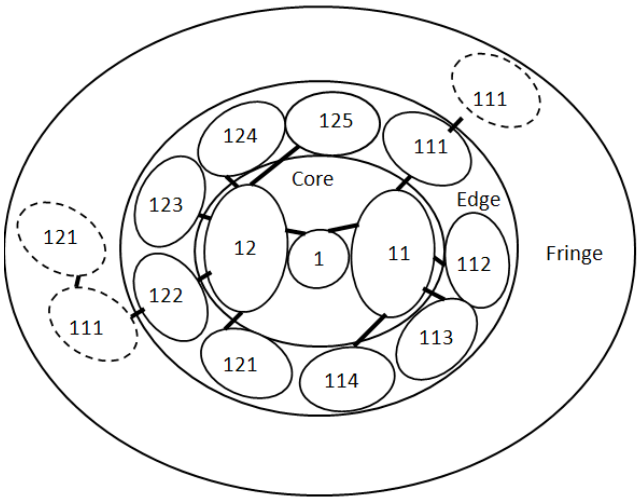


Figure 3.1: Networking prefixing

Network Address Translation (NAT) is a popular method to provide Internet access to devices that are mobile. NAT enables a node to receive a private address and enables the address to masquerade suing a local Public Address that is usually shared by all devices that have using NAT in that domain. In [44],the authors detail that the use of NAT is not a clean solution and propose the introduction of autonomous domains. Turfnet is their proposed mechanism to form individual autonomous networks, which can provide vertical connectivity with other autonomous turfs. Turfnet use gateways and a turf controller to establish and control the autonomous zone. Turfnet does not address the possible fragmentation of the autonomous domain, nor does it address concerns relating to the fragmentation of the address space.

Looking for a solution to the mobile Mesh problem, Chau et al. proposed Inter Domain Routing for MANET (IDRM) [45] as a first-step solution to the mobile autonomous system Mesh problem. Each IDRM autonomous system has a number of set gateway nodes that uses an internal routing protocol to maintain internal communication. Beacons are used to detect a split in the Mesh Ad-Hoc Network (MANET) autonomous system: if there is a split, a new MANET ID is randomly generated. Using a random algorithm for fragment ID-generation would leave fragments unable to independently calculate the new ID. Owing to the arbitrary partition, IDRM relies on communicating a membership digest of the nodes within the autonomous system rather than using the IP and prefix advertisement, as found on the Internet. IDRM uses a randomly generated autonomous ID to enable route documentation between autonomous systems. This approach would work for pockets of fragmentation, but would not scale to the use required by the future Internet. Core routing tables would be overwhelmed by the mass fragmentation on the fringe. But the separation of identity address provides some flexibility when dealing with the wireless domain.

The formation of autonomous domains and subsequent fragmentation has been discussed by Chau et al. [46]. Their proposal IDRM addresses many issues and provides a solution to the fragmentation problem through the use of broadcasts. IDRM uses a randomly generated autonomous ID to enable route documentation between autonomous systems; this approach would work for pockets of fragmentation but would not scale to the use required by the future Internet. Core routing tables would be overwhelmed by the mass fragmentation on the fringe. Using the DHT-inspired approach alongside the BGP-inspired approach, as suggested in [46],could overcome the scalability concerns. This would require an alternative approach to correspond with mobile node communication found in the current Internet.

Using a Distributed Hash Table (DHT) combined with the BGP-inspired approach suggested in [46] could overcome scalability concerns. Importantly, this would require an alternative approach to correspondent to mobile node communication found in the current Internet. Solving this challenge is an objective of this research project. One of the most important concepts relating to IDRM is the maintenance of autonomy within each Mesh. It is important that individual Mesh autonomous system operate their own internal security and routing policy whilst maintaining relationships with other equally autonomous network segments. Cooperation between autonomous systems enables reachability, and autonomy provides individual autonomous systems with the opportunity to create a secure network, running the appropriate routing protocol for their internal power/performance requirements.

Within the fringe, Internet mobile network fragments will cause the fracturing of the prefix that directs traffic to the current edge of the ISP network or point-of-fringe core interaction. The mass fragmentation on the fringe Internet that would occur at 8 am as many people go to work would result in heavy fragmentation and an intolerable extra burden on edge and core routers. Even without this extra burden, current BGP core routers are faced with tremendous growth of BGP tables [47]. The complexity introduced by the fragmentation on the fringe must not and would not be permitted to propagate to core Internet routers. In [48], the authors explore the potential of Multi-Protocol Label Switching (MPLS) to support the micro-mobility of mobile nodes, which could provide an alternative path between mobile node and correspondent, bypassing the Layer 3 aggregation. They do not look at using the scheme for macro-mobility or consider the issues relating to scaling the system beyond micro-mobility. It can be envisaged that increasing the quantity of exceptions at a single point would incur additional overhead when selecting and implementing a forwarding rule.

Each time a router receives a packet, it needs to compare the address against its own list of known destinations and accordingly forward the packet. The look-up process needs to be completed as quickly as possible so not to delay the routing of the packet. Owing to the increasing speed of interfaces and the resulting increase in the number of packets being transmitted, aggregating the look-up table is essential in terms of maintaining look-up speed. Applying individual routing entries to deal with mobility would increase routing table size to unmanageable levels. There would also be a constant requirement to recompile the routing table for each exception. In [49], the authors detail the requirements of the next generation Internet, noting the requirement of a split in the current scheme of combing location and device ID into a single address space, as found in IP. They also note that the advancement of computer virtualisation had been important in many areas, and that it could be as equally important when looking to solve the requirements of next-generation Internet.

Hierarchical Routing Architecture (HRA), as proposed by Xu et al. in [50], proposes a carrier/region/city approach to identifying the location of a device. As yet, the HRA scheme lacks the functionality to support mobile networks or nodes. But it does introduce an interesting concept of using a global DHT space to identify nodes. This is a concept that this work will build upon to create global indirection spaces.

3.1.2 DHT supported Network Mobility

The use of DHT as a next-generation Internet protocol replacement has been suggested by Hanka et al. in [51]. They ascertain that the current Internet has many shortcomings and will face more challenges when looking to future demands. They cite mobility as a critical area that needs to be addressed by any new approach, and the ability to freely roam between different networks establishing secure trusted relationships as a key requirement.

Hanka et al in [51] propose a redesign, which would accomplish their goals, through the utilisation of a DHT as a primary mechanism in achieving disassociation with the past hierarchical address partitioning used in the current Internet. They do, however, propose the use

of local temporary addresses—a move that is seen as a relic from the current Internet that should not be suggested for future protocol design. In [49], the authors discuss the possibility of a split between the address of location and identifier using overlay systems. DHT have also been suggested as a replacement for the Domain Name System in [52]. DHT schemes provide resilience in the face of network failure and are suitable for critical infrastructure application. As the DHT becomes fragmented, they can maintain their addressing scheme, and reachability can be maintained for those nodes on that DHT fragment.

Issues relating to the work of Hanka et al:

- They don't consider the sleep state of devices merely concentrating on the reachability of permanently active nodes.
- However, the use of DHT as a mechanism for storing name to identity and providing a mechanism for independent operation is something that we will take forward in this work.

Ekta [53] is a proposed protocol for use in mobile ad-hoc networks that uses a DHT overlay. Ekta shields many of the issues found in mobile networks, such as fault tolerance, scalability, availability, load-balancing and object location, which is seen as one of the more critical initial requirements in our protocol design. In [54], the authors detail that the initialisation of an overlay DHT in a mobile ad-hoc environment differs from those that exist on the Internet. They provide a solution in the form of Iterative Successor Pointer Rewiring Protocol (ISPRP), which directly applies a DHT over a Link Layer protocol using energy-constrained devices. Their work does not address the mobility of low-power Mesh networks and their attachment to the Internet. Moreover, in [55], the authors detail a DHT replacement to the Hybrid Wireless Mesh Routing Protocol (HWMP) found in 802.11s [18]. DHTs are suggested to replace HWMP to reduce the number of routing updates required. Their suggested scheme is aimed at a domain under single autonomous control and not individual autonomous Mesh networks.

3.2 Indirection

Internet protocol V4 and V6 uses a single address that combines both the location of the device and its identity at its point of attachment. This combination results in the tethering of the devices to the location at which the address was issued. In [56], the authors detail the use of global identifiers that correspondent nodes can use to communicate without the requirements of knowing the current location of a device; this permits correspondents to communicate using an identity that does not relate to their current point of attachment. This system has a number of benefits: firstly, transparent mobility; secondly, the ability for the correspondent to react to the traffic it is receiving; and thirdly, the ability to change its identity. This has been identified as a mechanism to defend against denial of service attacks, as shown by Adkins et al. in [57], where the benefits of implementing indirection in their architecture are detailed.

Critical infrastructure can often find themselves the subject of distributed denial of service attack. It is recognised that providing a mechanism to avoid such an attack would fit the critical infrastructure requirements of this project. Two of the main security benefits are the ability for

the host to hide their IP addresses and their ability to change their ID when they detect an attack. Unfortunately, the paper fails to mention that, if a host is hidden, this could provide a method of obscuring a hacker's identity. In [56], Menth et al. also detail the problems associated with the integration of IPv6 and IPv4 throughout the transition period. Loc/ID could provide a mechanism to interface between the two protocols.

In [58], Menth et al. detail the operation of Loc/ID schemes. They recognise that, although these schemes use an ID to locate the current point of attachment of a node, they revert to using the IP address currently assigned to the mobile host. They note that schemes that deviate from the use of IP after initial ID look-up are usually not compatible with current application software and protocol stacks. This is important as rapid acceptance is key to a protocol's future implementation. A protocol that requires the rewriting of current applications will certainly cause resistance from the software development community. Such approaches are usually termed 'clean slate' as they require a different routing and development approach. In [59], the author evaluates the current Internet and notes that the IP address is overloaded, and suggests a new approach—the host identity protocol (HIP). HIP introduces a separation in identifier and location using the IP address as the identifier and a new name-space to for the locator. In HIP, each host has a public and private key. The public key serves as the Host Identifier Tag (HIT). Each HIT is 128 bits long and generated using the SHA-1 hash function. HIP introduces a new layer between the transport and network layer. As opposed to using the IP address and port values to connect the host application, the application specifies the HIT or, if local, the local scope identifier. Converting the HIP to the current IP address of the correspondent should occur at the new layer between the transport and Internetworking layer. This would involve modifying the host's protocol stack whilst maintaining the original application. The current HIT of the correspondent can be retrieved using DNS. Notably, it is the responsibility of the mobile node to update the DNS with the new point of attachment—the IP address.

There are a number of problems associated with the use of the DNS to track the current location of the mobile node. This centres on the latency involved in updating the DNS and the information propagating the DNS. Using DNS would also result in in a single point of failure or attack, and could be used to identify the mobile host's current location. HIP specifies a four-way handshake between the sender and receiver so that both can be validated. Data transmission is then encapsulated into a tunnel between the endpoints. HIP provides an improvement in traditional IP through the verification of nodes. This additional handshake overhead could have adverse impacts on low-power mobile nodes. One of the key security features of indirection is removed by the HIP client, making direct contact with the IP address of the mobile node. Aspects of the key generation and initial handshaking could benefit the proposed scheme.

Indirection with a 128-bit random identifier poses a problem in the context of low-power mobile networks. DHT induces path stretch when communicating between nodes. This additional burden—which is not a significant issue to a desktop PC—poses a problem amongst low-power mobile nodes. Low-power transmission standards usually have small frame sizes, meaning that the use of a 128-bit address takes up valuable frame space. Traditionally, DHT operates on top of a Layer 3 protocol, encapsulating the DHT messages inside Layer 3, and Layer 2 frames would result in an unmanageable overhead. Using low-power reactive Mesh protocols coupled

with path stretch results in a number of broadcasts for sending a single DHT message. When running the DHT on top of a Layer 3 protocol, the DHT requires that all nodes are in a converged state and are globally reachable. Global reachability can be accomplished where convergence is not possible throughout the use of a translator, such as a NAT. When considering that a network with mobile Mesh networks cannot be in a converged state, an alternative to traditional P2P systems is then required.

In [60], Stoica et al. present an overlay-based Internet Indirection protocol (i3). The scheme decouples the act of transmitting and receiving through the deployment of an indirection layer based on top of the Chord DHT [61]. In the scheme, nodes do not communicate directly; instead, packets are placed at a point of indirection, decoupling the act of transmission and reception. With this noted, i3 is reliant on an underlying DHT, namely Chord. There is no reference to the preservation of the device state using the indirection interface; however, the scheme is not intended to support the Internet of Things.

Issues relating to current indirection approaches:

- Existing schemes dons specify support for low power IoT type devices.
- No specification for the management of Sleep state.
- Indirection point only used for the communication of state information.

3.3 Wireless Routing and Storage

1.1.1 Internet of Things

The Internet of Things (IoT) is seen as the next evolutionary step in the development of the Internet. It involves pushing internet protocols and services to the devices that are embedded within our environment [30]. Due to the low-power nature of IoT devices, networks of IoT devices must utilise low-power wireless mesh sensor networking technology. Typically, this involves deploying wireless Mesh routing protocols, such as DSR [15].There has been a great deal of research centred on the development of wireless Mesh sensor networks, and there are numerous bespoke protocols.

Initially, it was considered that the application of IP on constrained devices—which often run for years on a set of batteries and have limited processing, storage and bandwidth capabilities—was unrealistic. However, there have been an increasing number of implementations of IPv6 stacks targeting low-power sensors. 6LowPAN [32] is a low-power variant of IPv6 that utilises header compression so as to enable the transmission of IPv6 within the limited 802.15.4 link layer frame. Moreover, 6LowPAN provides the opportunity for every device in the Internet of Things to have a unique identifier. Coupling this identifier with the UDP CoAP [62] service that

is being developed by the CoRE working group provides a full service URI e.g. coap://fe80::202:b38e:ac13/pressure. Routing 6LowPAN packets between nodes is accomplished using Ipv6 routing protocol for low-power lossy links (RPL) [21].

Ipv6 on low-power devices will provide seamless integration with the wider Internet. However, it will not provide the robustness required within certain scenarios where there is the possibility of device loss. On the Internet, high-level protocols can be used to cope with these problems; nonetheless, the resource-constrained nature of devices in the IoT will impede the use of higher level services to aide with redundancy. Additionally, the wireless Mesh communication utilised in low-power wireless domains provides flexibility in cooperation that cannot be seen in the current Internet owing to its limited ability of physically connected devices to form new ad-hoc relationships. Normally, relationships on the Internet are formed more through long standing, and are based on firm commercial peering arrangements [63].

Issues with the current IoT standards:

- Porting web standards to IoT devices without addressing the low power sleep state that these devices find themselves.
- Current approaches don't address the mobility of devices.
- Don't consider the operation and reachability of devices if the network is separated from the Internet.

1.1.3 Wireless DHT

Hashing is a mechanism of mapping I information tuples into a smaller index *N*. Hashing functions are used to generate a key *K* that is $0 \leq K \leq N-1$. Hashing functions determine the distribution of *I* into *N*, where the relationship between *I* and *N* determines both the chance and quantity of collisions. A collision is the result of multiple *I* mapping to a single *N*. Hash maps usually implement what is commonly referred to as a bucket, which collects all instances of *I* that map to the same *N*. Usually, the bucket is implemented using liked lists. In order to avoid an imbalance between lists, hashing should provide a uniform distribution across *N*. Search time is proportional to the average number of keys in each *N*; however, this is dependent on the hash function, providing even distribution across *N*. Alternatively, a self-balancing binary search tree can be used to store tuples from the index; this will reduce the search time from *O(avgdepthN)* to *O(logavgdepthN)*. Distributed hash tables implement a hash space over a number of separate nodes.

Hash tables are used effectively in peer-to-peer systems where nodes frequently join and leave the network [61]. There have been a number of distributed hash table implementations, including Chord [61], Can [64], Pastry [47], tapestry[65], Kademlia [66]and Viceroy[67].

Chord uses consistent hashing, which is a technique that balances load as nodes roughly receive the same number of keys. Previous consistent hashing approaches have required each node to have knowledge of most other nodes, which presents an obstacle to scalability. Chord requires

each node to have information about *O(LogN)* nodes and can resolve using *O(Log N)* look-ups and routing overhead as *O(Log 2 N)*.

Distributed Hash Tables (DHTs) provide properties that are useful when considering redundancy in distributed fixed or wireless networks. DHTs can operate in a decentralised manner once an initial bootstrapping phase is completed [68] and providing that identities are allocated in a balanced way, the distribution of content should be even. The distribution of data is also dependent on the keying technique used to derive the new bounded identity [69]. Wireless networks can utilise DHTs in two main ways, either as overlay or underlay.

In wireless overlay DHTs, nodes initially have an IP address or other converged identity space with a DHT space operating on top. An example of this is Virtual Ring Routing [70], it uses a chord overlay to route between nodes. This introduces a 40% path stretch over the shortest path between nodes. This is problematic when considering the low power nature of the devices involved.

Protocols like Virtual Ring Routing can run a storage protocol running on top of the DHT. For example, in [71], Yang et al. propose the Ad-hoc Storage Overlay System (ASOS) for MANETS, describing the use of high-availability peers to store delay-tolerant data. They make the assumption that the use of a DHT on top of an existing scheme gives uniform distribution of data. The use of topology-independent structuring works well in environments with large node quantities with sufficient bandwidth and a converged lower routing layer. Sensor networks lack these properties and would result in the probability that data would not be distributed in some instances, but instead could be stored in the same locality.

Issues relating to Ad-hoc Storage Overlay System:

- Placement of data might not be uniform in the DHT space.
- It is likely that data might be placed in the same location when replicated in the overlay address space.

Underlay DHTs have no existing converged addressing space; there may be pre-existing identity schemes but nodes are usually unable to communicate directly using those identities. Typically, nodes are provided with identity using one of three methods, namely: central bootstrap, involving nodes being provided with identities using a breadth first or similar graph traversal, geographical position or distributed graph construction. In [72], the construction of a DHT based on the localisation of devices is proposed, and the centralised allocation of the identity space is suggested. The DART protocol [73] is an example of a distributed identity space construction. The use of geographical routing provides each node in the network with an identity related to its position in the network. This can be achieved through the use of a GPS-enabled sensor network or by applying identities on other position-approximation schemes [74]. The use of underlay DHT to accommodate data storage is not evaluated in these schemes.

Ensuring the preservation of a device's state, considering the failure of a percentage of the distributed storage system, will require that the information be stored in multiple locations. Storing multiple copies of the same information adds redundancy at the cost of storing the

additional copies. The number of copies and, more importantly, their placement will have a direct effect on the fault tolerance of the system. The following section will identify a mechanism enabling nodes to distribute a rendezvous state to a DHT based on a URI. Following the introduction of the data placement mechanism, a scheme evaluation is provided, as well a comparison of the use of the underlay position-relative DHT and overlay DHT under two network disturbance scenarios.

Application of DHT to ad-hoc networking include Kademlia-based ad-hoc routing (KDSR)[75], Dynamic Peer-to-peer Source Routing (DPSR) [76], EKTA [77], Iterative Successor Point Rewiring Protocol (ISPRP) [54], CrossRoad [78]and Virtual ring routing[79].

DHT provides the following characteristics that would support their use in critical infrastructure:

- Load Balance
- Decentralisation
- Scalability
- Availability
- Flexible Naming.

Ekta [77] is a proposed protocol for use in mobile ad-hoc networks that uses a DHT substrate because it shields many of the issues found in mobile network, such as fault tolerance, scalability availability, load-balancing, and locating objects, which is seen as one of the more critical initial requirements in out protocol design. In [54], the authors detail that the initialisation of an overlay DHT in a mobile network in a mobile ad-hoc environment differs from those that exist on the Internet. They provide a solution in the form of Iterative Successor Pointer Rewiring Protocol (ISPRP), which directly applies a DHT over a Link Layer protocol using energy-constrained devices.

Neither EKTA nor ISRRP evaluate their approaches to determine if they are capable of retaining information when dealing with the loss of individual nodes. This would be an essential requirement of any solution where the retention of information is required once the system starts to fial.

In [80],the authors note that scalable source routing combined with chord, such as hash-based routing, can achieve scalability with small per-node forwarding tables. This approach would be in keeping with the requirements found in devices attached to sensor networks.

DHT-OLSR[81] uses an amalgamation of DHT and OLSR within a single MANET to provide end-to-end connectivity. Using DHT-OLSR, a number of local nodes defined by a given number of hops runs a local OSLR instance. If the destination with which a node wishes to communicate is not found in the local table, a DHT based uni-cast forwards the packet to more remote nodes. The results found in [81] show that DHT-OLSR outperforms the protocols standardised by the IETF. DHT-OLSR use the hashed IP address of the node to create a virtual identity. This would cause the same issues as most protocols of this type that the identity is not related to the

position of the node so placement of data would be random in the topology, nodes would not be able to guarantee separation of data in the topology.

Policy is an important consideration when dealing with the interconnection of MANETs. In [46], the authors detail the following key aspect to domain autonomy:

- Topological relationship: Domains will peer and maintain relationship with stub domains.
- Dynamism: Relationships between domains will change the relationships between them.
- External Links: Managing the use of external resources, e.g. Satellite, WIFI and Mobile Telephony.
- QOS: Widely varying service issues with the likely requirement of carrying real-time importance, i.e. Health Informatics data.
- Per-Flow Policy: Critical operations might require adaptions in relationships to facilitate communication.
- Physical Characteristics: Determining the physical world and its associated impact on the topology and QOS state of the network.
- Strategic Deployment: Identifying areas of stability, such a vehicular ad-hoc networks or environmental sensing, which will provide staticity that should be utilised in topology formation.
- Security: Domains may not wish to route information though other domains. Alternatively, there may be a requirement to increase encryption at the service layer if such an event were to occur.

In [82], the authors details the importance of exploiting the proximity of networks when creating a distributed hash table. This would be especially important for any proposed mobility management protocol as to reduce the path length for messages propagating between mobile fragments. The address should be hashed so that the proximity to the fragments is maintained; this may require a more dynamic approach to the Layer 3.5 topology construction found in the DHT protocols detailed earlier.

3.4 Identity provision through localisation

All of the routing protocols examined thus far either utilise the IP/MAC/Random identity to perform a routing function or use an identity obtained from GPS. The networks distributed routing function provides intermediate nodes with route information so as to enable frame delivery. Relative addressing schemes, sometimes referred to as hierarchical addressing, provides an alternative forwarding mechanism that uses an address relative to the nodes' location in the network instead of the MAC or IP address. Usually, a tree data structure is applied, with packets routed in worst case via the root node to other branches in the tree.

DART [73] is a relative hierarchical routing protocol separating a node's identity from its address. The address is used to indicate a node's position within the network. The authors note that their scheme uses the address space efficiently when nodes are randomly, uniformly distributed and statically located.

DART provides three major functions: address allocation, routing, and address look-up. Address allocation maintains the address for each node in the tree that is relative to the node's position within the network. Routing performs the delivery of the frame from source to destination based on the relative address. Node look-up provides a mechanism to store the mapping of devices' ID to the relative address. Nodes join the network by listening to neighbours to locate an empty address location. Once a node identifies an empty position, it allocates itself the address, informing its neighbours of is acquisition. Subsequently, the node sends the hash of its ID and current address to the node that matches the address of the ID hashed. Addresses in DART are l bit binary numbers, and the address space can also be viewed as a tree of l+1 levels. The leaves of the tree represent the addresses of nodes in the network. Notably, nodes that are close to one another should form a sub-graph of the tree. Each node maintains a list of its siblings so as to enable the completion of forwarding decision-based destinations relative to the address of the frame.

The current address allocation scheme in DART does not consider balancing the relative addressing tree, nor does it provide a mechanism to enable the protocol to select the optimal number of bits for the number of nodes to be used by the network. Maintaining an appropriate address length will help reduce frame transmission overhead; this will be important when considering the small frame size used in 802.15.4 networks. When considering mobility, future protocols would need to minimise the impact of updating the new node's position to the overlay.

Issues relating to DART:

- There is no indication of how the address space would balance.
- It is not intended to be a storage protocol.
- Does not consider Internet scale reachability.

The prediction of future state could provide additional information to help with the address space balancing issue. The analysis of the current and previous network state to plan for future capacity would provide a better estimation as to future address space requirements. In order to implement such a scheme, knowledge of the relative locations of the devices would be required. Localisation-aware systems can utilise GPS [83] with the aim of identifying node location; however, this is problematic as this places reliance on the availability of the GPS signal. In highly built-up areas, coordinates provided by GPS are known to wander. When considering the use of GPS in critical infrastructure, some countries would be concerned that the US government—which controls the system—could disrupt a GPS-reliant system. Force directed graphs [84] have been applied to provide an approximation of relative location in sensor networks [85][86].

Issues relating to GPS based approaches:

- Devices must have a GPS sensor to obtain an identity.
- Increased Bill of Materials for those devices.
- Additional power consumption
- Line of sight required with satellites
- GPS system can be switched off by the US government.

As previously mentioned, DHT provides a useful abstraction to facilitate reliable and robust data dissemination in wireless sensor networks. There are three main approaches to building a DHT in wireless sensor networks: 1) Overlay DHT, where the address space is built on top of an existing converged protocol [87]; 2) Virtual, where nodes construct a tree [88]; and 3) physical location-based schemes [89], which map N dimensional spaces onto the actual or estimated physical location of the node. Tree routing schemes provide a single dimension address space. This could result in nodes that are distant in the identity space residing in close proximity in the physical space. Overlay DHT do not provide any mechanism for insuring the placement of data is separated in the topology.

For example in [90], Awad et al. propose a virtual location scheme, referred to as Virtual Chord Protocol (VCP). In the scheme, nodes are provided with an identity within the range spanning 0–1.As nodes connect, they obtain an identity relating to that of a neighbour; this identity is used to route packets, as well as identity keys, that map to that node. Such a scheme has the capacity to create unbalanced address spaces. This is owing to the nature of the address allocation: as nodes join the system, the existing address space at a particular location will be partitioned. Depending on the ordering of nodes joining the system, you would have areas of the address space that have been heavily partitioned and areas where there has been little partitioning; this results in data items being disproportionately placed at areas of low partitioning, resulting in an imbalance. Scatterpastry [87] can be implemented using either overlay or underlay DHT. When using the overlay mechanism, there must be an existing Layer 2 frame-forwarding mechanism in place on the network, such as Destination-Sequenced Distance Vector (DSDV), for example[91]. This requires that the transmission in the DHT space cause path stretch in the Layer 2 space.

Geographic Hash Tables, as proposed by Ratnasamy et al. in [92], detail the Data Centric Storage where user data is pushed into an identity space formed by the physical real-world coordinates obtained via GPS sensors. This type of scheme can be useful when the accuracy of placement is essential; however, the strictness of identity related to a physical real-world position relies on the good physical distribution of nodes throughout the coordinate space to combat imbalance. An alternative approach is to use localisation in an effort to estimate the position of the nodes. This does not improve balancing but does remove the requirement for GPS sensors. It is not realistic for this work to consider using schemes that utilise hardware based location systems.

3.4.1 Localisation

Localisation algorithms provide a mechanism to identify the positions of individual nodes within wireless networks. Such information can then be provided to the individual devices so they can make forwarding decisions. Depending on the network deployment requirements, it might be possible to equip a subset of the nodes with a GPS or physically record their location. This provides valuable information when attempting to identify the location of the remaining nodes. These schemes are referred to as anchor-based localisation. In [93], nodes use the hop-based position estimate from a GPS-enabled device, which leads to a decentralised system, it is reliant on the external GPS system on some nodes and the continued operation of the subset of devices equipped with GPS receivers.

Anchor-free localisation does not require information external to the sensor network; instead, it utilises the information from the sensor network with the aim of estimating the relative positioning of devices. Usually, individual devices transmit the information they hold about the network back to a central co-ordinating unit. The information passed could include the following: Neighbour Identities, Node Identity, Signal Strength of incoming packets (RSSI), bit error rates, and ultrasonic/temperature or other sensory information. This information can be used by the localisation algorithm with the aim of determining the location of the devices. Owing to the high search space, it is common for approaches to use probabilistic Meta heuristics. For example, in [94], Chagas et al. apply Genetic Algorithms and Simulated annealing using RSSI values from sensors to identify their location. Other schemes make use of graph-drawing algorithms and produce good results [85], typically using Kamada Kawai or Fruchterman-Reingold. Kamada Kawai utilises spring force [84], whereas Fruchterman-Reingold uses a force directed algorithm [95].

In [96], Nawaz et al. detail an anchor-free localisation mechanism that utilises a modified graph-drawing algorithm. The approach is based on the Kamada Kawai graph drawing algorithm [84], utilising a sensor equipped with range-finding devices. This approach provides a good estimation of device location. It does not provide a balanced topology.

Genetic Algorithms (GAs) have also been used for localisation. Genetic algorithms are a type of search heuristic that model a living organisms evolution. In [97], Zhang et al. detail the implementation of a GA to identify a node's position in a bounded two-dimensional space. The mutation of individual node position is bound by their current location, and the fitness function rewards the correct placement of nodes with respect to their neighbours. The use of genetic algorithms to create a candidate topology takes longer than the graph based approaches.

GHT networks require an identity that provides a node with the opportunity to conduct greedy forwarding. If we are to use the GHT for distributed data storage, we need the address space to be evenly distributed across the nodes in the topology. If the protocol was to use GPS sensors, the network would be bound to those identities, meaning that, in order to obtain even data distribution, it would be required that the physical nodes be positioned in a grid layout, which would be unrealistic in most scenarios. Alternatively, the scheme could use localisation techniques; however, these are geared to provide a good estimate of the physical location of the node, replicating the issues with GPS. A scheme is required to allocate addresses that are relatively positioned to maintain reachability but which are distant enough to provide equal coverage of the two-dimensional bounds of the DHT, providing a balanced distribution of data between all nodes.

3.5 Geographical Hash Table Routing

There have been many studies investigating the ability of decentralised GHT routing protocols to mitigate the impact of the local minimum that occurs in wireless sensor networks. It is

important that when reviewing the existing schemes that they are evaluated against the requirements of this thesis namely both routing and storage of data. It is not sufficient that the protocol just escapes local minimum, it is also required that the data be placed in a location that provides a fair balance. The work in this area is rooted in geometric graph traversal problems, an early example of which is Compass Routing on Geometric Networks [98]. This scheme utilises Delaunay Triangulations to enable nodes to perform face routing. Face routing protocols traverse the interior faces limited by those links whose edge crosses with the line between source and destination vertices. This scheme would permit packets to traverse the topology but would not balance the placement of data if there was a partial loss of the system. There is no mechanism to detect the failure within the topology. Greedy Perimeter Stateless Routing (GPSR) [99] is a widely used example of a protocol that utilises the right-hand rule on a planar graph. In GPSR, nodes send periodic beacons that include their locations in a predefined 2D space. Nodes route data using this one-hop neighbour information, forwarding data to a node that is geometrically closer to the destination. In Figure 3.2, a greedy scheme forwarding data from node A to E sees A assuming it is the closest to E as its neighbours B and D have a lager geometric distance. Again this approach does not provide a mechanism to balance data placement with topologies that have experienced failure.

GPSR uses the right-hand rule to identify nodes that are located on the void on a route between the previous node and the destination. It achieves this by identifying a neighbour node whose coordinate is the first counter-clockwise in the adjacency list.

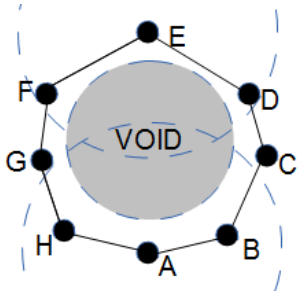


Figure 3.2: Perimeter Routing

This process is continued until a neighbour is identified as being closer to the destination node. GPSR causes data to route along the border of the fault, which causes a routing imbalance. It also fails to detail the way in which data should be handled if the information is destined to be stored in the fault. It does not describe when data should be stored on a border node if the target node is in the failure. If data saves to a border node, this would cause the border nodes to take the storage responsibility for all of the nodes that have failed within the fault.

Both Compass Routing on Geometric Networks [98] and Greedy Perimeter Stateless Routing (GPSR) [99] can operate in an entirely distributed way. Obviously, each node requires a coordinate to be allocated to each device in the network.

Carbuna et al. in [100] describe the use of Voronoi diagrams for the allocation of regions of the sub-graph to an individual node in each Voronoi cell, whose connectivity supports it. As stated

in [100], such an approach requires large computation requirements that would be too great for the devices that operate in this space to accommodate, placing a requirement on a centralised management scheme. If it is desired that an implementation be robust in the face of failure, a scheme requiring a centrally controlled response would be too slow to respond to changes in the topology, which would be further exacerbated if a network was to be under continuous failure.

The approaches detailed above provide mechanisms for forwarding information as close to the destination as possible, with little regard for the impact of the traversal on the sensor network lifespan. Usually, devices that operate in this space are energy-constrained, maximising the lifespan of this system at least to the point where it has completed its objective is an important area or work. In [101], Renda et al. propose load-balancing in Load Balancing Hashing in Geographic Hash Tables. They detail an alternative approach to changing the existing geo-routing protocol that, in most cases, relies on computing complex geometric transformations. Instead, they propose modifying the hash function that is used to store data in the GHT. They propose two approaches: the first uses a destination density function; the second is a heuristic approach that operates when assumptions fail. In [102], Sakar et al. describe an approach to balancing storage and traffic load through the application of Mobius transforms and Ricci Flow algorithms to create covering spaces to mitigate the impact of topology irregularities. The approach is computationally expensive; however, they note that small changes to the topology can be made with only a small overhead.

In [103], Gao et al. detail a clean approach to avoiding the imbalance that occurs on the edge of hole. They implement a round robin approach where the hole's perimeter expands and then contracts, creating larger virtual holes. This would have the disadvantage of creating larger paths but would reduce the load on hole edge nodes. This approach would share the routing overhead a limited distance around the fault. For large faults this might just overwhelm the additional nodes that are sharing the workload. The scheme would also fail to balance the data that is intended to rest inside the fault. The use of neighbour load is utilised in Geographic Load Balanced Routing (GLBR)[104] as a mechanism to avoid overloaded paths. Nodes consider the overhead effecting neighbour nodes before making greedy forwarding decisions. This could lead to devices making different forwarding decisions leading to nodes placing data at a location in the topology that is not repeatable. This would result in corresponding nodes not finding the data one placed. Curveball Routing [105] is another approach that projects paths onto an alternative surface Curveball routes on virtual coordinates obtained by projecting the network on a sphere. This is primarily a mechanism to remap the DHT space fairly to a set of nodes to avoid the disproportionate amount of traffic that gets routed through the centre of the topology. The scheme does not show how it would deal with the loss of sections of the topology. It would require global communication of the hash mechanism. It does provide a mechanism that we will look to extend in this work. Instead of using a global shifting scheme, this work will look to create a local change to the topology to mitigate the impact of failure.

The schemes we have surveyed either require global knowledge to compute an appropriate hashing algorithm to balance the topology or utilize a fault traversal mechanism that operate when a packet encounters local minimum. Utilizing global knowledge to create initial topologies is acceptable. However, attempting to flood the topology at a time of failure would

result in a reduction in available bandwidth, at a time where the network should be distributing systems state as opposed to forwarding global network information. If we wait until we encounter local minimum before finding an alternative route would remove some of the storage imbalance but retain the forwarding imbalance. Data intended for inside the fault would still reside on a node exhibiting local minimum and might also result in less than optimal traversal around the fault.

3.6 Summary

This chapter surveyed the work relating to the management of mobile devices and the management of routing and data storage in wireless sensor networks utilising DHT technology. The review identified that existing mobility schemes do not consider the reachability of low power wireless devices when they are connected to wireless mesh networks. Existing schemes also fail to deal with the sleep state of these devices. The review highlights the possibilities of extending the work of indirection architectures to provide support for the reachability and sleep state devices through the application of indirection. In these schemes the DHT serves as a mechanism for routing and also a storage mechanism to map the identity and current DHT identity of a node. Information can also be stored in the DHT such as the devices current state. This could prove useful if a device is lost due to fire or other disruptive incident. The review identified literature relating to the use of DHT in mesh networks. These schemes do not look to integrate with the global reachability required or deal with the imbalance that can be caused when utilising Geographical Hash Tables, either during initialisation or following the failure of a section of the network. Schemes do provide geometrical techniques to avoid failure. This would be adequate for routing but not storage.

The next chapter will describe IoMANETs—an architecture to provide global reachability to mobile and static Internet of Things devices.

Chapter 4

Mobility, Routing and State Redundancy for the Future Internet of Things

This chapter introduces the Indirection Overlay for MANETs (IoMANETs) Architecture. IoMANETs provides a fault-tolerant, scalable solution to the Internet of Things machine-to-machine communication problem. This chapter presents a protocol capable of supporting the requirements of Future Cities scenarios that are outlined in the context section of this Chapter. It is the intention that this architecture may be deployed in isolation, notably without the need for Future City interfaces. This chapter will evaluate the design and identify the main challenges to the implementation. These challenges will be explored in later chapters. IoMANETs provides a globally reachable identity to both fixed and wireless devices, i.e. devices can communicate directly with any other device; this should not be discriminatory. Moreover, a mobile sensing device will be able to communicate with a wired device and vice versa.

Machine-to-machine (M2M) low-power mobile devices and mobile Mesh networks will become ubiquitous in the future Internet of Things. Mobile networks will need to interact with static networks so as to ensure continuous connectivity with the core Internet. Maintaining device connectivity whilst mobile will provide increased capability and opportunity for M2M interaction. In turn, this will provide greater fidelity to higher-level decision-making systems, such as a Future Cities platform. It is considered a primary design goal that the protocol has the capability to manage Mobile Wireless Mesh Networks: for example, a wireless network that is located in a building that is on fire would need to interface with the wireless networks of the first responders. It is also possible that these networks could become fragmented during their operation, leading to an inter-connectivity requirement between fragments.

This chapter begins by identifying the context for this work and outlines the high-level system designs, required to manage the real-time analysis, visualisation and modification of a failing systems within a Smart City. The approach detailed in the rest of this thesis aims to minimise the

impact of system failure upon critical infrastructure and to identify the requirements of low-level routing protocols to support the application of Internet of Things technology to critical infrastructure.

This chapter makes contributions to the mobility management of Low Power Wireless devices and Networks as well as the management and preservation of state mitigating the impact of device sleep state and management.

4.1 Future Cities Context

This section details the context that will affect the design of the IoMANETs architecture and influence the design of wireless protocols in Chapters 5, 6, and 7. This section introduces Future Cities platforms, with focus directed towards the sub-systems required to protect critical infrastructure, utilising the Sensor Actuator capabilities of the Future Internet of Things.

3.2.1 Smart City, Internet of Things Integration

Future Cities Critical Infrastructure response will coexist within the Future Cities ecosystem and will use information that will be provided through the deployment of Internet of Things technology. One of the important aspects of this work is the provision of an aggregation management interface for Future Cities (Figure 4.1). This will result in Cities consuming data from Low Power Internet of Things IP endpoints. These devices might be fixed of mobile.

It is envisaged that each component of a city will utilise a data aggregation service as the primary mechanism for data ingress from external data sources. This will provide annotated data streams from systems and system collections, such as Internet of Things Networks. Each data aggregation point will attach to the Future Cities platform and will subscribe to a cross-cutting concern service. Cross-cutting concerns can be considered as the functionality provided by the Future City Platform. Data will help cities react to cross cutting concerns.

Examples of cross-cutting concerns are as follows:

- **Smart Grid**

  Devices would report energy consumption and take part in the demand side response. This would enable a Future City to meet its power requirements by controlling the power use of individual devices.

- **Governance/City Operations**

  Future Cities will operate more efficiently responding to the requirements of its citizens. Rivers can have their levels automatically checked. Bins can be identified as being full. Street lighting can be automated.

- **Health**

  Body Area networks can provide information in real-time relating to wellbeing. If an elderly person falls in their home, an ambulance can be dispatched automatically.

- **Emergency Response**

  From providing an initial notification of an incident, such as a fire alarm, through to the endpoints providing additional reporting from devices in the vicinity of the failure.

Under normal operating conditions, a system would provide low overhead update messages for each concern. Informative updates could be transmitted to a concern, such as a surplus energy advertisement available at a reduced unit cost. This would be processed by the Future City platform that would, in turn, produce a response. In the case of a power advertisement systems could inform devices on the network of the opportunity to consume surplus power at a reduced rate. In the result of an incident, a concern would provide information concerning the incident escalation.

As an example, if a smoke detector goes off in a residential property, initially, this might not be deemed a serious problem as fire alarms go off from time to time. As it becomes apparent that there is a problem, the Smart City would request additional information, such as the status of all fire alarms and possibly the status of neighbouring fire alarms. As additional computation becomes necessary, the Smart City could provide additional cloud IaaS capacity (critical response instance). It is important that aggregators perform local reasoning to avoid overburdening high-level systems. The aggregator should also be capable of storing local historical data. The aggregator might be located in a cloud service.

In the following sections, the requirements for future cities will be provided.

3.2.2 Maintenance and Extraction of Failing SAN State

Systems operating within a permitted tolerance range will provide information through the usual Smart City aggregation points. Following a partial system failure, this communication flow may get disrupted. In this case, sensor actuator networks should cooperate to enable the continued operation of a system and provide a mechanism whereby responders can extract state and invoke change. It is essential that the current state of a device, as well as the previous state of devices that may have been destroyed, is accessible to the response system.

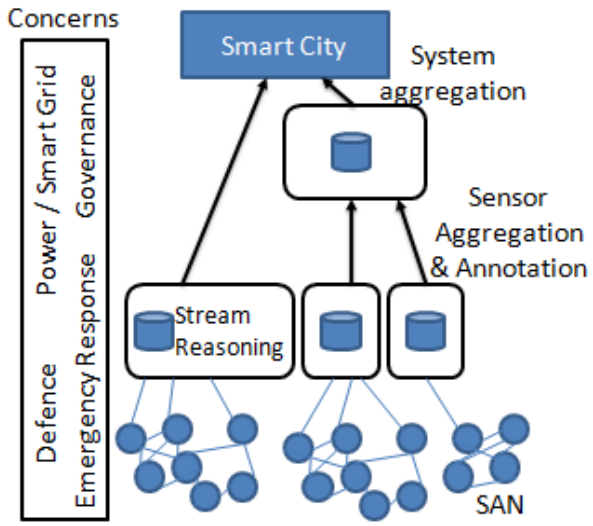


Figure 4.1: Smart City Information Flow showing Cross-cutting Concerns

It is common in post-failure scenarios that forensic investigation is carried out to determine the cause of a failure. In a complex system, this is difficult and may prove impossible if a system has become damaged as a result of the incident. The state of an individual sensor and its possible destruction could also be important to minimise the impact of an ongoing failure to a system. It is important that future protocol development enables the protection of sensor state within a failing sensor actuator network.

In an effort to fulfil the requirements of fault-tolerant access to a sensor actuator network, a distributed sensor state service that can operate once the network has entered the failed state is required.

3.2.3 Critical Infrastructure Visualisation

It is important that real-time information is provided to first responders in a way that is easy to understand and which provides them with mechanisms to drill down into a system so that changes can be made and propagated into the failing system. There will be physical restraints as to the type of visualisation that would be possible. In some circumstances, it might only be feasible to send a text message or use a pager. At the other extreme, however, there could be mobile response vehicles equipped with high-power computer systems. For this reason, the system described in this work would need to accommodate a range of interface technologies. So as to avoid creating interfaces for a range of devices, it would be best to collect devices into specific groups and to provide generic interfaces, such as simplex text, duplex 3D and an indication of bandwidth, as well as local processing capability.

Based on the identified Critical Infrastructure response requirements, four main system components have been identified.

- Smart Cities Systems Annotation and aggregation Service
- Critical Response Reasoning Instance
- Critical Response Visualisation and Control
- Sensor Actuator Network Overlay State Management.

Figure 4.2 shows an example scenario that future cities platform would need to manage. The diagram shows three systems of which a single system number (3) is a member of a critical infrastructure. Systems1 and 3 provide data via their usual aggregation points; System 2 is failing and can no longer provide information. The Smart City has created a reasoning instance and provides the required plans and real-time data streams. First responders access the SAN, providing data to the Reasoning instance, which, in turn, provides response suggestions. Emergency Responders are then able to interact with the failing system.

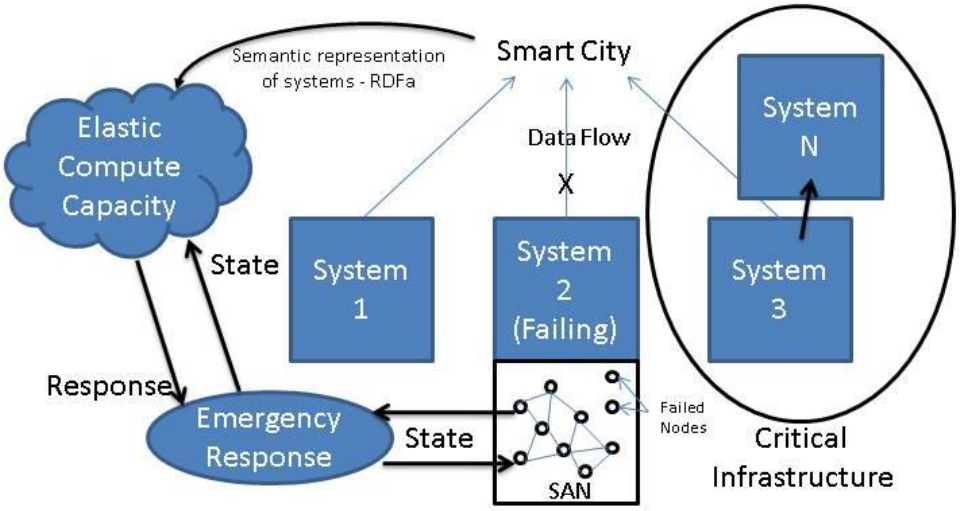


Figure 4.2: Future City Scenario

3.2.1 Smart Cities Systems Annotation and Aggregation Service

Objects, such as lights, sensors, ovens and pumps, for example—all of which are a part of individual systems, namely factories and homes, require that IoT devices are connected to the Internet and have global identities. Each system will be equipped with an aggregator that can communicate with cross-cutting concerns, e.g. power and governance emergency service. The aggregator will provide local storage for information streams and provide annotation.

3.2.2 Critical Response Reasoning Instance

Initiated by the Smart City, a response instance will use the information provided to enable it to reason about the current situation. Information must be annotated in an effort to enable automated reasoning using appropriate ontology.

3.2.3 Critical Response Visualisation and Control

Using a three-dimensional interface, Critical Infrastructure View will provide a detailed representation of the failing system, detailing recommendations from the Critical Response Reasoning Instance. The interface will enable the user to interface with the SAN of the failing system

3.2.4 Sensor Actuator Network Overlay State Management

SAN devices that are unable to contact their aggregation provider and have lost contact with the PLC that would usually set their state will form a structured overlay so as to enable the distribution and preservation of state. Figure 4.3 shows a failing san. Using the system representation provided by the city to the response instance, individual devices can be accessed as well as the previous states of destroyed devices. This information can be used to help form an well as the previous.

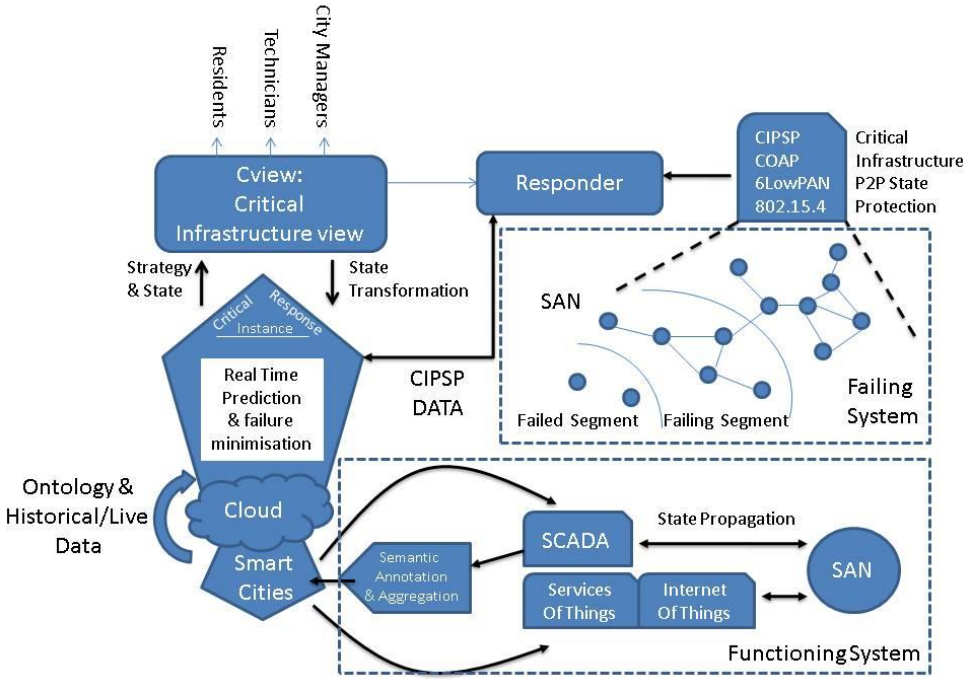


Figure 4.3: System Level Framework diagram

To support data acquisition for future cities services, architectures are required that will support devices to communicate their state whilst they are connected to the Internet and also when they

are mobile. Importantly the device should be able preserve state within the current pervasive environment, to protect the information that the devices holds in case of destruction.

4.2 Objective

The objective of this chapter is to detail an Architecture that will operate within the constraints imposed by the low power devices that will be deployed to the Internet of Things. The Architecture will support mobile devices and device collections maintain connectivity with correspondents on the Internet, providing mechanism to protect their distributed state if they are destroyed.

The review of related work in chapter 3 identified the following issues:

- Current Indirection schemes do not consider the management of Low Power Sensors connected to wireless Mesh Networks.
- Wireless DHT schemes have been proposed as standalone solutions and are not integrated with global identity schemes.
- There are no mechanisms currently provided to maintain a device state to support destruction or sleep state within the area or indirection support mobility.

To alleviate the issue of reachability of individual Mobile Internet of Things devices and device collections, the application of a global indirection scheme that provides a mechanism for a mobile node to receive a new identity at its current point of attachment is suggested. This will be defined by its position relative to other autonomous system that are connected to a router that intersects the fringe and edge Internet. In order to enable a correspondent to communicate with the mobile node using this new identity, the mobile node hashes its IP and registers its current ID with this hash in the overlay. Correspondents can then address packets to the ID and have the overlay deliver the packets to the mobile node. If a mobile node feels that its current state is important to the overall effectiveness of a distributed application, it can submit its state to the overlay. This could be in preparation of a sleep state or may be seen in the event that destruction is likely.

Maintaining the addressing and autonomy of fringe Internet mobile networks with the added complication of fragmentation is complex; nevertheless, there are other factors that need to be considered. Many of these mobile devices will be constrained in regards to processing and battery power. Autonomous systems or fragments of autonomous systems may sleep at any time and for any duration. Sleeping nodes may awaken to find that either their own or surrounding autonomous systems' mobility has had an impact on their routing table. Once awake, the autonomous system must go to great expense to first detect changes and then take the necessary actions to converge. The network could then transmit a small amount of data before going back to sleep. In this case, a disproportionate amount of the device capacity has been allocated to network management function. Any scheme that was developed to support the fragmentation and mobility of the autonomous system would also need to cope with the constraints of those devices.

The review of the literature provided in this thesis identifies the use of indirection to break the link between identity and location. Utilising peer-to-peer overlay technology, combined with indirection, provides a robust mechanism to support device mobility. The distributed nature of the overlay would provide the scalability as well as fault-tolerant properties required of critical infrastructure.

Nodes located at their usual point of attachment will be reachable via an assigned Internet Protocol address. If the routing information is in a converged state, the IP address will also provide the location of the device; this will enable a remote corresponding node to communicate with that node. When a node moves from that point, it will need to maintain the identity provided by the IP address; however, the location provided by that identity will be lost. Data destined for that node will be routed to the usual point of attachment. Altering the usual flow of packets destined for the node is central to managing the mobility problem. The packets should be routed to a node that is responsible for the current connectivity of the destination mobile node, or to a device that has been given responsibility for the software state of a device. IoMANETs will provide the facility to break the link between location and identifier through indirection

4.3 Contribution - The IOMANET Architecture Overview

IoMANETs' primary purpose is to provide a mechanism to enable communication between application processes running on separate mobile nodes or between fixed and mobile nodes. It is assumed that the fixed node is connected to the Internet with either IPv4 or IPv6 whilst the mobile node has an *802.15.4* adapter, operating a 6LowPAN IP stack. The interconnectivity between the edge and fringe Internet would be performed by an IPv6/6LowPAN router, using an appropriate translator, such as a Jackdaw network card, for example. Within the suggested scheme, the device is referred to as an Edge Router. IoMANETs have been designed to be protocol-agnostic; however, the initial reference implementation initially supports IPv4 and IPv6 to the static client and IPv6 to the mobile low-power node. The application of IPv6 is essential as devices that operate within the Internet of Things are expected to run a low-power variant of IPv6 stack called 6LowPAN. This is owing to the quantity of devices expected to operate in this domain and the limited IPv4 address space.

A single 6LowPAN domain has an assigned global IPv6 network range, with each mobile device calculating its own address based on the combination of its MAC and network address advertised by the Edge Router. Each 6LowPAN domain comprises two node types: static nodes (not limited by power availability) and a number of battery-powered devices that must conserve their energy by entering a sleep state. Battery-powered devices could leave their own 6LowPAN domain and need to attach to a second 6LowPAN domain.

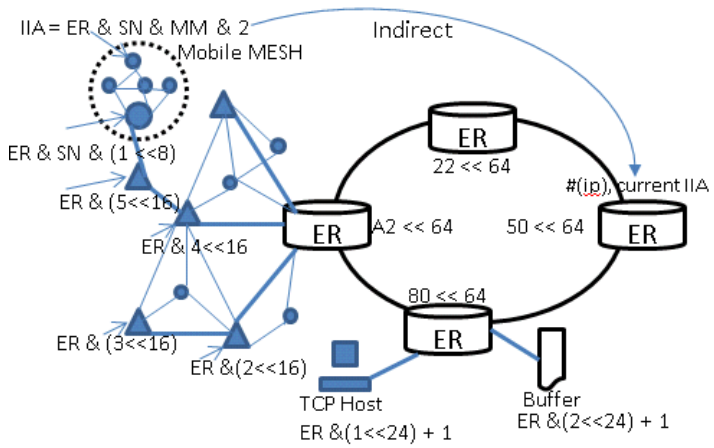


Figure 4.4: IoMANETs Physical Topology

When a device attaches to a 6LowPAN domain that is not its home network, it is only able to communicate using its link local address, as a global address would not be allocated. IoMANETs facilitate the reachability of the device using indirection based on the original global address. Figure 4.4 shows four Edge Routers that have formed a single IoMANET instance. The bottom Edge Router has a single static node and a buffer that is maintained for a mobile device. The Edge router on the left-hand side has a 6LowPAN network with a number of nodes; however, only the nodes represented as triangles have been accepted as part of the mobility scheme. The mobile Mesh network, denoted by the dotted circle, contains a mobile Mesh network that has been associated with a stable host Mesh node. Nodes within the Mesh are then able to perform indirection to the physical location to which they are currently attached. Alternatively, a node can indirect to a buffer that can be accessed when the node next associates.

IoMANETs is based on two distributed P2P topologies, providing the following functions:

- The storage of indirection information containing current point of attachment address or the location to store frames until a mobile host becomes active.
- A mechanism to store frames on behalf of a sleeping node.
- A routing function to deliver packets to either a static or mobile node.
- The provision of a temporary identity to a mobile node that relates to their point of attachment.

IoMANETs can be divided into two main operational components, described in the following subsections. This section provides a detailed description of the IoMANETs framework.

4.3.1 Mobility overlay Supporting Indirection and Packet Delivery (MoSIPD)

MoSIPD provides an overlay network to support the indirection of packets to mobile Internet of Things devices. Edge Routers (ER) attached to the Internet form a MoSIPD instance, which can be either private or public. MoSIPD makes use of the Chord DHT as an overlay provisioning new Indirection Identity Addresses (IIA) for mobile nodes or mobile MANETs, and providing a routing mechanism to that address. Edge Routers that are connected to the Internet and which want to take part in the IoMANETs MoSIPD first must identify an ER that is part of the MoSIPD instance, which can be completed using the following bootstrap process:

- Static Configuration (manually configured through pre-arranged agreements)
- Previous neighbour relationships
- Broadcast indirection overlay join requests to directly connected neighbours
- Contact dedicated look-up service

Once in contact with the MoSIPD instance, the ER creates a random IIA for use in the overlay network. This address is 128 bits long, with each ER using bits 128 through to 65 inclusive. The first 64 bits are used with the aim of providing local identifiers to mobile nodes/Mesh that are not part of the ER network but which have joined the relative addressing mobility tree overlay through association with a node that has sufficient resources. Addresses can also be assigned to static nodes that need to communicate with mobile nodes that are attached to the MoSIPD instance. Addresses may also be used to label frame buffers, used by nodes with only intermittent connectivity. The 128-bit IIA address space is detailed in Figure 4.5, whilst the formation of this relative addressing tree is detailed in the next section.

Edge Routers receive traffic through the Chord overlay, and either forward it based on the Chord routing table or accept it for processing. If the packet is an indirection key set request `i[#(myip),myIIA]`, the key is stored and distributed to neighbour Chord Edge Routers for redundancy. The ER could then receive requests for the current IIA associated with a key `ikr[#(ip),returnIIA]`. These are returned to the requesting node through the overlay. The requesting node can then transmit data through the overlay to the target IIA `data [srcIIA,dstIIA,ib,length,data]`. Nodes about to enter a sleep state may request an IIA from an ER to be associated with a temporary storage buffer`sbr [RND,IIA]`. Upon confirmation, the requesting node updates its indirection key within the overlay to redirect to the temporary buffer. The buffer confirmation also contains a randomly generated token that must be submitted to receive the data contained in the buffer. When data frames are received, they are then forwarded to Chord neighbours, stored in a buffer or accepted for delivery. If the packet is to be transmitted to a node connected to the ER, the TCP connection table is checked to determine whether the node is listed. This table enumerates static nodes' addresses that are communicating with mobile nodes within the IoMANETs overlay. Packets are forwarded to the TCP socket associated with the IIA. If an identifier is listed in the Wireless Mobility Border Protocol (WMBP) table, the frame is transferred to that sub-system for delivery. If the packet is destined for a temporary buffer, the frame is added to the bottom of the linked list associated

with that address. The MoSIPD sub-system monitors key set commands to keep track of mobile nodes within its domain.

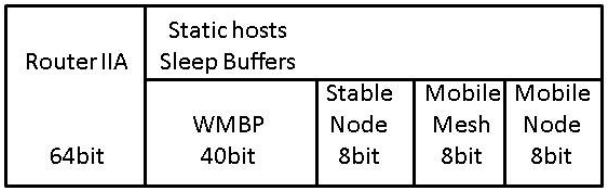


Figure 4.5: IoMANETs Address Space

4.3.2 Wireless Mobility Border Protocol (WMBP)

Single nodes or MANETs use internal WMBP to establish internal state and external WMBP to identify neighbours and accordingly configure a relative addressing tree. This is not a requirement and WMBP can use alternative structures. In chapter 5 a 2 dimensional GHT systems is described to enable nodes to redundantly save information within the wireless domain. It is the responsibility of WMBP to forward overlay frames to the root, thus providing connectivity to the global indirection architecture, or otherwise to route data between attached nodes.

The following subsections describe the use of external and internal WMBP.

a) External WMBP

Stable nodes able to provide assistance to mobile nodes create a 32-bit random identifier and start listening for UDP multicast frames. The Edge Router, by default, is the root node in the relative addressing overlay, and starts broadcasting UDP messages to its neighbours. Nodes seeking to join the overlay receive the frame, taking note of the hop count, incoming signal power and the source address of the frame. The node increments the hop count value and retransmits the root advertisement. This broadcast floods the network and results in each node having a list of potential overlay neighbours, along with their relative power values and the shortest path back to the root node. This stabilisation period continues for 20 root broadcasts.

Each node transmits their neighbour information to the root node `mtn[rid,nc,{neighRID,power}..n]`. The root node creates a minimum depth spanning tree, where nodes are assigned addresses using a breadth first walk of the tree.On the return traversal, the node's maximum degree is recorded. Those at the bottom of the tree have max degree 0 and the root node's degree is calculated as `stable_node_mask + host_mask`. The root node utilises the power map to identify the relative position of each of the nodes in the network using a localisation technique, this is unspecified within the framework, however, an approach that utilises genetic algorithms to create a balanced topology in is detailed in Section 5. The topology can be used to predict future frame delivery points. The root node

then sends the allocated addresses' depth first using the temporary ID of the mobile node as a target `mta[iia,rid,degree]`.

Once nodes have been assigned an ID, they can route using the relative addressing scheme. If a node receives a message addressed with an ID in the range [node_id, max_degree], it consults its neighbour table to identify the host with the next minimum value match. The data will then be forwarded to that node; this process is repeated until a stable node matches the stable address. Stable nodes provide random identities to mobile gateway nodes that request association. This process is maintained by the Internal WMBP service, described in the next section. Figure 4.6 shows a Mesh network (top image), with each node comprising a random identity. The bottom image shows the minimum spanning tree with each node assigned a WMBP identity.

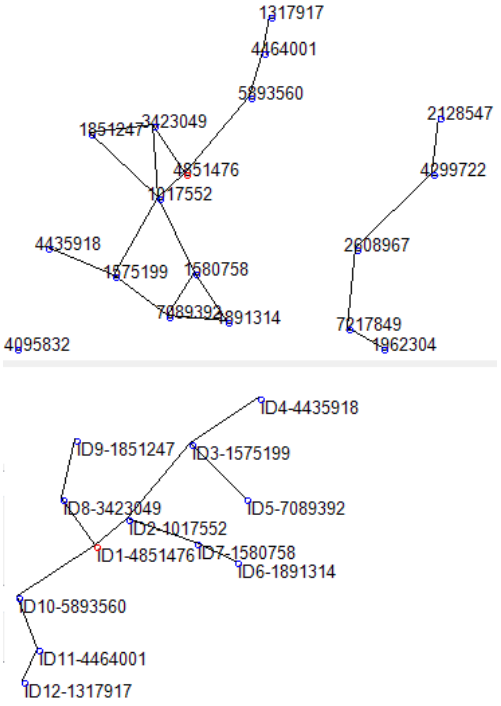


Figure 4.6: IoMANETs Simulator Output

b) Internal WMBP

Mobile nodes sharing the same global prefix elect an internal node to act as a gateway. Usually, when dealing with a body area network, only a single device would have the capability to transmit to a stable node in the host network. Mobile nodes that are a part of the mobile Mesh transmit an `indirect_when_able[#(ip)]` packet to the gateway. When the gateway is provided with an IIA range from a stable node in the host network, it can then transmit the Indirection requests for all nodes that have made a request. Once confirmed, the mobile node can interact with the overlay using its IIA.

4.4 Evaluation

The evaluation of this framework applies a simulated environment to assess the performance of the initialisation protocol and the capability of the architecture to cope with the limitations of the wireless interface. The following sections outline the simulation environment and results.

4.4.1 Simulation Testbed

In order to evaluate the proposed architecture and ensure that the scheme can operate within the constrained resources of low-power devices, a 6LowPAN test bed has been constructed to validate the designs. The network comprises eight Atmel AVR raven development boards divided between two 6LowPAN networks. Jackdaw *802.15.4* interfaces are used to route between the wireless Mesh and virtual machine instances. These are interconnected using an IaaS cloud of 30 nodes, representative of Edge Routers. The Atmel motes use the Contiki operating system.

4.4.2 Evaluation of the Approach

We test the IoMANETs architecture using an urban network scenario with pedestrian mobility patterns. Experimentation shows that the reception of the Atmel Raven mote provides an average coverage radius of 12.5 metres. Walking a straight trip through the coverage area results in a 20% packet loss. Sending 40 UDP data packets typically results in 23 successful acknowledgements. IoMANETs is capable of associating with the host network, requesting two storage buffers, establishing indirection to those buffers and performing a single read on each buffer within the given constrains.

4.5 Conclusion

The investigations carried out have provided real-world constraints imposed by mobile 802.15.4 devices running the 6LowPAN IP stack. The implementation of a hybrid overlay provides a robust scalable topology that works within the constraints imposed by the air interface, whilst at the same time taking advantage of the fast interconnection between Edge Routers. The approach detailed in this chapter to indirect traffic to a temporary buffer overcomes the reachability issues associated with devices that exhibit intermittent connectivity owing to mobility or sleep state. The application of a distributed indirection system provides a mechanism to locate a device without the requirement of a single home agent.

In its current form IoMANETs would not be suitable for use in vehicular networks. The problems associated with fast mobility are inherent to relative addressing schemes, such as those used to underpin WMBP. In an attempt to mitigate these effects, route prediction mechanisms could be integrated into IoMANETs, operating as follows:

- During the WMBP stabilisation period, the Edge Router receives a power map from each of the nodes in its broadcast range.
- Using mobile node association patterns, association prediction can be provided, enabling mobile nodes to register their indirection information at time *t* upon entry to the network.
- Any data arriving at time *t+n* will be forwarded to the WMBP mobile Mesh portion appended to the predicted stable node address.
- The buffering system detailed in Section 3 (to include state transition) can also be extended.

Indirection provides a mechanism to alter the flow of traffic intended for a specific mobile application instance. It would be beneficial, in some scenarios, for an application to submit its current state information to the overlay, which would enable a correspondent to communicate and alter the last submitted state of a mobile application. Additional information would need to be included so as to inform the correspondent of the ability of the current state to affect change upon its environment. This could be used to help resolve the deadlock that occurs when devices have alternative sleep patterns, or to provide an artefact in the event that a node is destroyed. This would be useful in critical infrastructure where post-failure analysis is essential to improving long-term system viability.

The next chapter will extend the work associated with the Mobility Border Protocol Domain Tree (MDT) to provide a distribution mechanism to allow devices to communicate using indirection in the wireless domain. This is to fulfil the requirement identified in Chapter 3 where it is established that the Internet of Things devices should have the capability to communicate with each other in the event of the failure of the main Internet connection.

The protocol detailed in the next chapter will include a mechanism that retains information for devices that may have failed permanently or have been destroyed.

Chapter 5

Rendezvous Redundancy for the Internet of Things

The Internet of Things will result in the connection of billions of embedded devices to the Internet, which will provide a connected substrate that will power our economy and provide greater control and sensing within our environment. Low-power processors and wireless devices will be connected to the objects with which individuals will interact every day. These devices will communicate with home and industrial automation systems to provide optimal configuration and operation within their environment. Devices will cooperate outside of these zones, with individual homes and organisations coalescing to form smart neighbourhoods and cities. Concerns that override local needs, such as energy management, which will be managed by smart grids, will place external dependencies on the availability of Internet of Things systems to interoperate.

Most devices attached to the Internet of Things will operate in an autonomic fashion. They will attempt to survive within the limited communication conditions in which they are deployed. In order to preserve their longevity, devices that are battery-powered will sleep and wake sporadically with the objectives to interact with their environment and to communicate. Devices attached to mains power would be more active and provide stable routes to central base stations.

Under normal operating conditions, devices will have identities allocated to them, commonly IPv6, resulting in application services being made available through the utilisation of universal resource identities, enabling distant systems to communicate. Devices will be expected to use these identifiers for machine-to-machine communication. Typically, this commutation will be between devices and a local management platform/aggregator. It is expected that devices will also communicate directly. This direct machine-to-machine (M2M) communication can cause

problems when taking into account the sleep state of devices and the possibility of permanent failure as a result of the environmental situation, such as a fire, flood or explosion.

The evaluation of a system post-failure will be an important concern for Internet of Things smart spaces. For example, after a home fire, investigators use techniques to identify the cause. The state of individual appliances in the home before, during and after the fire would provide valuable information to investigators. During the fire, first responders could identify the location of people in the building or the state of hazardous objects, such as gas bottles, for example. It would also be useful for first responders to be able to access the information of individual devices that have failed to help blue light services respond more efficiently. Importantly, Internet of Things devices must be able to operate when their connection to Internet has been compromised.

Providing a mechanism to enable machine-to-machine communication and state preservation whilst a system is failing is an essential requirement when considering the operational requirements of the Internet of Things.

This chapter details the operation of a data placement mechanism for GHT, and provides an evaluation of the application of position relative identity spaces and distributed hash tables when creating a distributed space to store redundant rendezvous states. This chapter also evaluates the capability of redundancy key schemes to preserve state within failing topologies.

This chapter makes contributions through the development of a new mechanism to geographically separate the placement of data.

5.1 Chapter Objective

The objective of this chapter is to detail a mechanism that will replicate data with Geographical Hash Tables so to ensure its advisability in the face of network failure. The Architecture in the previous Chapter outlines a mechanism whereby nodes can distribute their state within a global overlay. This counteracts the issues relating to the sleep state of mobile devices and preserves the state of devices. However if a network is disconnected from the Internet there is no mechanism to support any further redundancy. In this chapter we will introduce a mechanism that will operate using a Geographical Hash table to distribute data with geospatial separation to help preserve the state of devices. We will compare this work to the existing schemes that operate using overlay DHT.

The review of related work in chapter 3 identified the following issues:

- Overlay DHT provide a mechanism to distribute information with a key space. However the overlay cannot guarantee the separation of data. Leading to the possibility that all copies of the distributed information reside in the same area of the network.
- There is no comparison between the use of GHT for the redundant keying of information and overlay DHT.

The literature review in Chapter 3 identified the use of Overlay DHT such as Ekta [53] to manage the distribution of information within wireless sensor networks. Nodes that are operating using this type of protocol generate random identities in a 1 dimensional addressing space. This can result in nodes that are geographically close having distant identities. Placing data into these topologies with either random placement of where placement separated in the identity space does not change the likelihood that the data would rest on adjacent nodes. GHT schemes provide a n dimensional identity that is related to the nodes physical position in the network. Any logical distance in the key space will result in the physical separation of data in the network. The following section will detail the contribution of this work.

5.2 Contribution - A Rendezvous Redundancy Mechanism

This section will detail the operation of a novel rendezvous data placement mechanism for GHT to support the redundant distribution of data within GHT networks.

The subsequent section will provide an overview of the mechanism and will detail a keying mechanism to be used within the scheme so as to provide a redundant rendezvous communication primitive. In Section 5.4, an evaluation of the scheme against two failure models is provided.

5.2.1 Mechanism Overview

We define a wireless Mesh network N containing a number of wireless devices V that have wireless connections E forming an interconnected Mesh topology N . At least one device in N will be interconnected to a wired network providing Internet Edge Routing functionality.

$$N = G(V, E)$$

Equation 5.1: Network Definition

There will be a number of devices in the network with higher capability characteristics. These relate to battery power longevity, processing capability, storage capacity and network link quality. High capability devices will be assigned identities that are relative to their geospatial location within the network, mapped to a two-dimensional coordinate. Nodes are provided the identity through a central coordination function. The details of the function fall outside of the scope of this chapter but will be discussed in Chapter 6. All high-capability nodes in the network have a responsibility to forward and accordingly store the state of other nodes within the network if they have such a capacity.

Upon receiving a packet, nodes either forward the received packet or store the information locally contained in the packet. Each node achieves this by examining its own neighbour table and the destination of the packet. If there exists a node that is closer to the destination

coordinate than its own coordinate, the packet is then forwarded to that node. This process is repeated until a node receives the packet and its own identifier is the closest to the destination, at which point the node stores the state in its own internal memory. This type of routing is referred to as 'greedy forwarding'. Nodes cooperating in this manner create a distributed storage function, as shown in Equation 5.2.

$$g \subseteq V$$

Where $g_{xy} \in N_{0\,..xmax\,,0\,..ymax}$

Equation 5.2: Distribution Function

The size of the dimensions $nmax$ of the network is related to the total key space requirements; this will be a function of the number of nodes and the required key collision probability requirements. It is important to consider the payload requirements of the small 802.15.4 frames used to transmit data, balancing the key length requirement with anticipated payload size.

Initial experiments will use a 32-bit composite key of the 16-bit *x* and *y* coordinates. The exact choice of key size would vary depending on the deployment characteristics.

Each device will have its own limited internal memory containing the current state of the device and the state of those devices whose keyed state maps to the identity allocated to the device.

In a conventional distributed system, an individual device's internal state is commonly pushed or polled to a second device with the goal of the network of devices reaching an optimal global decision than they would if relying on an isolated world view. Devices communicate with each other by calling a web service function, or, as is found in Programmable Logic Controllers, individual memory words are pushed using iso tsap messages to remote devices.

This scheme provides a distributed rendezvous storage model, where the device state is proactively pushed into the distributed storage layer by the source node, as can be seen in Figure 5.1. In order to provide redundancy, a device is able to create a number of keys for each individual state, providing multiple copies of the same data stored at different locations in the network. As long as the communicating devices can independently calculate those keys, the remote devices can retrieve an individual copy of the source device state form the distributed storage layer.

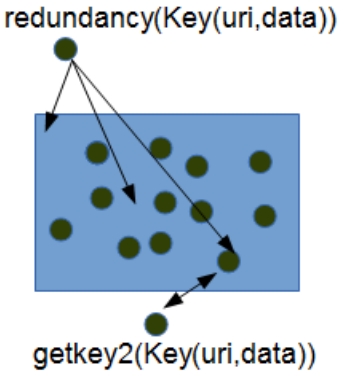


Figure 5.1: Redundant Rendezvous

For example, an alarm would compute N keys with the following input function key('http://10.11.9.8/alarm',true). Using a redundancy keying scheme known to all communicating devices the corresponding node can perform the same keying technique and request the data. In the example a single True/False state is used as a key. Additional meta information could also be stored relating to time stamps etc. enabling the validation of the data that is being returned. To reduce the overhead of the distributed retrieval of state data, devices could order the rendezvous points according to distance, retrieving the data from the node whose key is closest to its own. If a response is not forthcoming from a device each key can be queried in turn, until a state is returned or the list is exhausted. The effective distribution of the data to ensure its resiliency is linked to two key criteria; identity distribution and the redundancy keying technique.

Distributed hash spaces usually run on top of a converged routing space where nodes have an identity at Level 3 of the OSI model and a virtual distributed ID that is maintained at a higher level. Running a distributed structure on top of a structured topology, the identities would be randomly placed; this would result in the random placement of keys within that topology. Any attempt to force the separation in rendezvous points would be limited to the probability distribution of the distance between those points in random space.

The next section will define a function that will provide a mechanism for multiple key generations that guarantees separation.

5.3 Redundancy Keying

This section will detail a novel mechanism to transforming the tuple(uri,data) into a set of redundant geospatial keys that exhibit defined separation characteristics when applied to a position relative distributed storage space. In order to validate the capability of the system to retain individual device state, experiments will evaluate the use of uniform distance separation between keys of the same set shown in Equation 5.3.

$$p \subseteq V \supseteq g$$

$$\forall x\,[(x \in p).key[0..n] \rightarrow x \in k]$$

Equation 5.3: Redundant Keying

We define $k$as a set of keys generated by nodes P. P Are the nodes in V that have the capability to submit their state into the distributed rendezvous layer. The quantity of k will be dependent on the quantity of P and the number of keys that the redundancy function generates. g are nodes in V that get information from the rendezvous layer, a subset of P and g will get and put into the distributed rendezvous layer.

$$K = f(P)$$

$$P_n = \sum_{n=0}^{P} \sum_{k=0}^{q-1} (P_k)_{\substack{md5(uri)\%xmax + \frac{xmax}{dist} * q\%xmax, \\ md5(uri)\%ymax + \frac{ymax}{dist} * q\%ymax}}$$

Equation 5.4: Key Rotation

Nodes must be able to generate keys using a function that has the following properties:

- Independently repeatable using uri.
- Provide even distribution across N.
- Minimise the probability of the total loss of a nodes state.

Communicating devices must be capable of generating the first key and subsequent $N \rightarrow \propto$ keys of the function key(uri). Individual nodes can make an independent assessment of$N$; this may change due to dependence on the autonomic assessment of perceived threat, both to the individual node and the distributed storage layer. The generated keyset must provide an even distribution across the key space; if not, individual devices would become overwhelmed by the storage and routing requirements placed upon them. Essentially, the set of keys generated must be placed in a way that ensures the preservation of device state when the distribute rendezvous layer experiences failure owing to the loss of individual nodes or the deterioration of communication links between nodes.

The equation shown in 5.4 details the approach adopted to resolve the issues identified. For each node P, the key space is divided by the number of keys q to be generated by each individual node *P*. Subsequently, the MD5 cryptographic hash function is applied to the URI so as to

generate a base key that is shifted by the Nth portion of the identity space. This results in key tuple separated N times. The MD5 cryptographic hash of the URI is repeatable and provides excellent distribution. This initial point is then rotated providing N distributed keys with same level of uniqueness but are still repeatable in their generation.

5.4 Evaluation

This section will evaluate the Rendezvous Keying mechanism outlined in the previous section. To conduct the simulation we have designed a custom discreet event simulator that has been implemented in python. The simulator permits the evaluation of the key placement mechanism utilising a greedy forwarding wireless mesh routing protocol, with individual nodes allocated identities that are relative to their position in the topology. The simulator places 100 nodes arranged evenly in a grid formation. The simulator generates a total storage requirement for 30,000 nodes. Once the individual states have been saved, utilising the distribution mechanism combined with greedy forwarding, the simulation applies two failure patterns to test the capability of the mechanism to persist states in the face of network/node failure.

The two failure patterns are as follows:

- Gradient - Nodes are removed in stages from left to right in vertical strips.
- Radial - Nodes are removed in stages from the perimeter of the network to the network centre.

The goal for this experiment is to establish whether there is an advantage to utilising a position relative topology for the distributed storage of distance enforced key value pairs. This will be evaluated against the random distribution of data to represent the storage of key value pairs in an overlay topology. The Results of the simulation are provided in figure 5.2 and 5.3. Figure 5.2 shows the results of the gradient fault pattern with Figure 5.3 showing the results of the radial fault pattern. Both sets of results show the simulator incrementally removing 10% of the network from 10% to 90% on the x axis, with the total number of node state loss shown on the y axis.

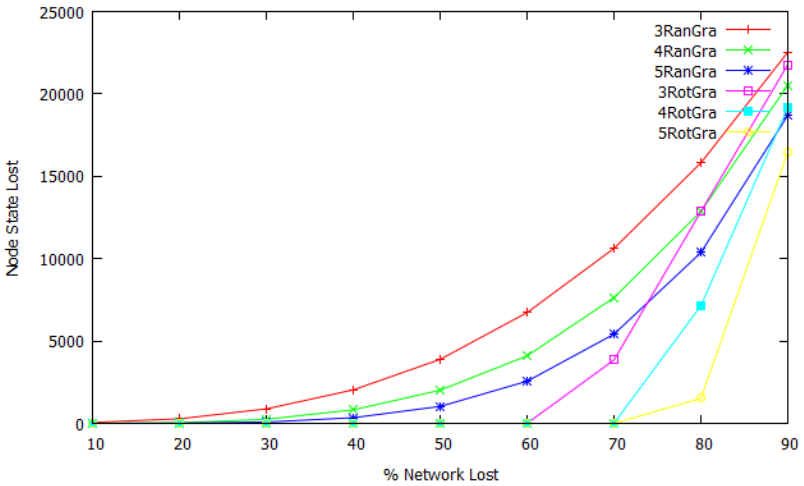


Figure 5.2: Random and Rotational node loss with R gradient fault pattern.

The simulation for the random and rotation node loss is run with the generation of 2, 4 and 5 keys for each node state. This is shown in the key for random generation as 3Ran, 4Ran and 5Ran. Rotational placement is shown as 3Rot, 4Rot and 5Rot. In both tables these labels are appended with Gra for gradient and Rot for rotational key loss. As a reminder the key rotation mechanism will generate multiple copies of each state and place them in geospatially distinct regions. Whereas the random placement will simulate the use of an overlay DHT, placing states randomly in the topology.

For both experiments we see that the random distribution of device state results in the total loss of state for individual nodes from the 20% failure point, this continues to rise proportionately with the percentage of network lost. Increasing the quantity of keys results in a reduction of the probability of the total loss of state, but it does not eradicate the loss. For example, when using three keys, the network loses ~4000 node states when 50% capacity is lost. When using 4 keys we can lose 60% of the network to encounter a similar loss and when using 5 keys we can sustain a 70% loss of network to achieve a similar loss in node states.

When applying the keying mechanism to the gradient fault pattern, we don't lose a device state until 60% of the network is lost, at this point using three keys results in the loss of 4000 states. At 80 percent loss we start to see the missing states for 4 and 5 key placement.

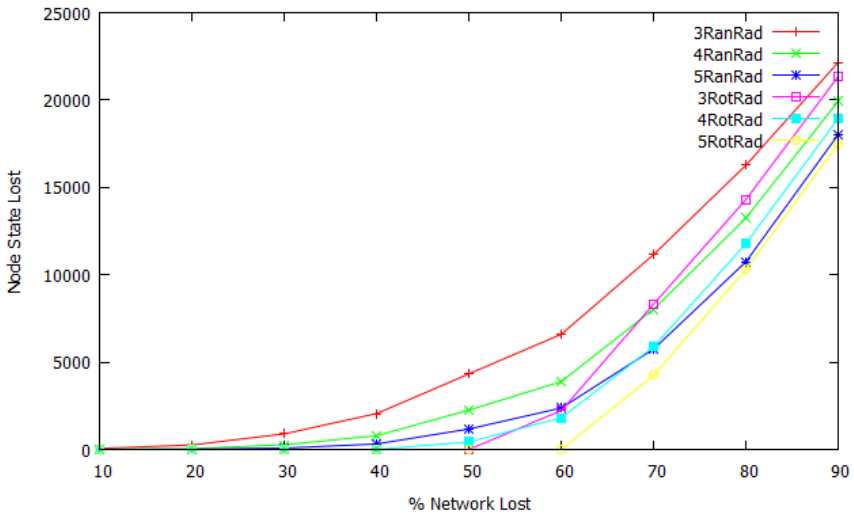


Figure 5.3: Random and Rotational node loss with Radial fault pattern.

When applying the radial loss pattern we see the same distribution of loss for the random placement of keys, this is expected and helps to validate the experiment. The radial loss pattern has an impact on the key rotational placement strategy. We start to see loss earlier, at 60 percent as opposed to the 70 found in the gradient loss. This relates to the modular operation that wraps the key around the topology when the generated key overruns the bounds of the topology.

The results detailed in figure 5.2 and 5.3 prove the assumptions relating to the loss of a devices state, when that state is placed at random positions within the network. If a node is to randomly place data within the overlay then there is the opportunity for every copy of the devices state to be placed within a region that is removed in the first iteration of the simulation, in both radial and gradient scenarios, leading to a total loss of device state. Following subsequent iterations of the simulation, the probability that every individual device state will be lost increases proportionally with the quantity of the network lost. The loss of state using random placement would be the similar in profile to if the topology was an unstructured overlay DHT with or without the application of distance keying.

The results shown in Figure 5.2 and 5.3 show that random placement follows the probability distribution of the state distance, highlighting the total loss of an individual devices state from early on (10%) in both the gradient and radial loss scenarios. In the gradient loss scenario, the findings show that the rotational key total loss is related to the probability of the key falling into the last remaining shift zone that is dictated by the number of keys generated and the size of the key space. Altering the number of keys that are rotated reduces the impact of extreme loss; however, this would have an adverse effect on the cost of communication. It could be envisaged that devices would select a key that would fit their importance in line with the distributed system goal.

When evaluating the impact of the radial system loss model, the results show that the effect of the random distribution is identical as expected; however, the radial distribution of 5 keys shows similar loss to the radial distribution of three keys in the gradient loss model. Importantly, the 5-key radial still outperforms the random placement of state.

5.5 Conclusion

This chapter has detailed a mechanism to distribute state within a GHT supporting the Internet of Things. The solution presented proposes a novel distribution mechanism for separating data items geo-spatially within a GHT.

The results emphasise that utilising a position relative addressing scheme with distance keying provides a better fault tolerance than random key placement when applying gradient and radial system failure models. It has been found that additional research is required in order to achieve comparable results when applying the radial fault scenarios to the key rotation mechanism.

The Internet of Things will enable the next generation of smart spaces, providing sensing and actuation capabilities to everyday objects both at home and in the workplace. It is important that these devices are able to function when the network is not operating at its optimum. Utilising existing Internet standards and those that are simply adapted to take account of the limited resource will not provide IoT devices with a robust communication mechanism to support post-failure analysis or support the communication of sate during or following an event that disrupts the system. The rendezvous communication mechanism detailed in this chapter provides devices with the capability of robustly distributing information, making it less susceptible to network destruction.

The following chapter will describe a mechanism for the provision of suitable addresses through localisation in environments where nodes are not evenly distributed across the topology.

Chapter 6

Balanced GHT Localisation using Evolutionary Algorithms

Identity is a key requirement in any network. It enables a device to be contacted and for the remote device to respond. In wireless mesh networks, devices not only perform the functions of endpoints but also enable the forwarding of information between endpoints. This function is commonly referred to as a 'routing'. In the context of fixed wired networks, the routing function is commonly performed by dedicated devices; however, in wireless networks, devices must cooperate to enable global reachability. When all nodes in the network are able to communicate with one another, the network is said to have converged.

In order to facilitate convergence in a wireless multi-hop network, devices must run a routing process, which runs on each wireless device and provides a mechanism for that device to construct a local routing data structure. These local data structures enable a device to make forwarding decisions that result in a message being passed closer to its intended destination. The forwarding decision is centred on a routing algorithm that operates on the data structure held by the individual device. In wireless multi-hop networks, information is forwarded from device to device until it reaches its final destination (a detailed review of wireless routing is provided in Chapter 2). This communication mechanism relies on the client (originator) establishing the identity of the destination (server), as well as the capability of the devices at intermediate points in the network, in addition to the devices' sufficient knowledge of topology to forward the packets that make up the information flow between communicating nodes.

Chapter 4 identified the requirements and architecture for routing within the wireless domain of the Internet of Things. One of the key requirements relates to the way in which nodes interact with each other when connected to the Internet, as well as how they operate at times when the wireless domain is failing. Wireless Internet of Things devices will be deployed within homes, industrial buildings and public spaces. Devices must be able to operate to some degree during a

systems failure. At the point of initialisation, nodes will cooperate with each other and a gateway that will connect them to a global distributed identity space. This approach, referred to as IoMANETs, is detailed in Chapter 4. This identity space will provide a distributed data structure so as to enable rendezvous-based communication. It was determined that rendezvous-based communication would provide a better communication primitive to ensure communication between devices that exhibit characteristics that mean their constant availability cannot be guaranteed. Rendezvous communication utilises a push–pull approach; this facilitates communication between devices. Markedly, devices that wish to communicate submit the payload to a third-party storage node. The destination device must poll the storage node to retrieve the information. Such communication has a greater cost than that of direct communication; however, it provides the benefit that source and destination do not need to be active at the same time; to enable communication, devices must be provided with an identity to enable communication.

We define an infrastructure wireless Mesh network as a connected graph with the following properties; that is, a graph constructed of nodes distributed in a two-dimensional coordinate space, whose identity is bound by the key space available.

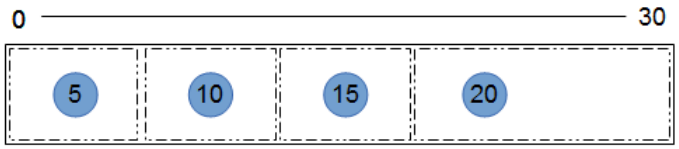


Figure 6.1: Address Space Imbalance

When allocating nodes with an identity within the available identity, it is important that the proportion of the address space that maps to that node is proportional. Figure 6.1 shows a simple example of a one-dimensional address space. Due to the offset of the nodes, Device 20 is assigned a disproportionate portion of the address space.

This chapter will detail a mechanism to provide nodes in a wireless network with an identity to enable greedy forwarding. The identity will relate to the nodes' position in the network due the application of the distributed rendezvous communication primitive discussed in the previous chapter. If node identities do not follow the physical topology, we will not benefit from the distribution of states that enable satisfactory redundancy in the face of device failure.

This chapter contributes an address allocation algorithm that provides balanced identity spaces for wireless Geographical Hash Tables.

6.1 Chapter Objective

The objective of this chapter is to address the issues concerning the generation of node identities to enable balanced GHT wireless topologies. The literature review in Chapter 3 identified that existing approaches concentrate on generating identities that resemble the physical placement of devices. This is not in itself a problem if the scheme is used purely for direct communication. However if the GHT is to be used as a storage layer then the topology could become imbalanced. This has the effect of overloading nodes and in the case of failure, losing a greater percentage of the saved information if the failure occurs in a portion of the network that has a disproportionate data allocation. The use of Genetic algorithms has been suggested to perform localisation but these too are affected by imbalance. This work will look to advance the genetic algorithm approach to generate balanced topologies.

The review of related work in chapter 3 identified the following issues:

- The current work relating to the localisation of GHT topologies concentrates on creating topologies that reflect the physical position of nodes in the topology.
- The current work relating to the application of genetic algorithms to creating topologies is limited to identifying placement. These approaches do not perform as well as the graph drawing approaches.

This work investigates the balancing problem inherent to the use of GHT address spaces in wireless sensor networks. In GHT, a node's identity relates to its physical position—either to a common reference, such as GPS, or simply relative to its position in regard to its neighbours [89]. Relating a node's position to its identity in the DHT provides additional resilience when storing information owing to the fact that distance in the virtual address space also results in the physical separation of data within the sensor network. Distributing information between a set of autonomous peers provides a number of benefits over pre-planned orchestrated data distribution schemes. One important property is the even distribution of data between the collaborating nodes. If data is not distributed evenly, there is a risk of overloading individual nodes storage and processing capabilities. The risk of data loss due to node failure also increases.

DHT are usually constructed above existing converged identity spaces. On the Internet, devices have an established identifier—commonly an IP address—allocated to the device by an upstream service provider. Devices on the Internet can use this identifier to achieve global reachability, providing the opportunity for connected nodes to construct an overlay DHT. Nodes wishing to join a DHT generate a random identity and contact a node that is already a member of the DHT to initiate the joining process. The discovery of a DHT and the joining process is usually referred to as 'bootstrapping'. The process of nodes joining with random identifiers should provide an even distribution of identities in the key space, resulting in nodes taking responsibility of an even portion of the identifier space.

The key space is usually an integer of a predefined bit length. It is important that there is sufficient space so that nodes can pick an identity at random with a low probability of collision.

It is also important that the keyspace is sufficiently large so that data elements that are keyed into the space do not collide. Upon the node joining the DHT overlay, devices create neighbour relationships with other nodes in the overlay; these are the only nodes that a device will contact directly using the underlying protocol, e.g. TCP/IP. Instead, messages are routed between neighbours of the overlay using the distance of the key to the neighbour entry as a method of forwarding data closer to its destination. This process can introduce path stretch [106] as nodes bypass the topological view of the supporting network and instead forward information based on the overlay topology. It is this property that makes overlay-based distributed hash tables unsuitable for sensor network deployments; however, there is ongoing work to make adaptations to improve the use of Overlay protocols in wireless environments [87].

As an alternative to the traditional overlay-based DHT found on the Internet, wireless sensor networks can utilise position relative identity spaces. In such schemes, nodes are provided with an identity relative to their physical location and/or other relative position metric, e.g. hop distance. A device can use its own identifier and those allocated to its neighbours to route information closer to a destination node—a process referred to as 'greedy forwarding', as noted earlier. Devices can use the same mechanism to distribute information into the GHT, as is found in regular DHT; nonetheless, when using GHT, there is no underlying service that can be used to directly identify a device; rather, this information must be stored in the overlay. This is commonly referred to as a distributed indirection point, requiring communicating devices to first lookup the current position-relative identity using the hash of the known identity. This relies on the node with which you want to communicate, having already performed the same hash function on its own identity and stored this in the overlay with its current address as the data element.

Before a device can take part in a GHT, it must first be associated with an identity that falls within the addressable range of the network. The network must be bounded; in GHT, this is usually the x and y coordinates between 0 and N, where N is dependent on the key length. If the devices in the network are keying data using a consistent hash function, such as MD5 on unique data, this should see the even distribution of keys in the coordinate space. If there is an even distribution of nodes in the coordinate space—either through physical location or virtual coordinate—then there is a good probability that each node will have an equal proportion of the keys to be stored. For nodes to communicate, they need to be aware of their position so they can assume the necessary identity, enabling them to forward and receive information. Commonly, schemes point to the use of GPS [77], although that has complications due to cost and the nature of deployment. The main alternative to using GPS or other physical locating scheme is to use Localisation.

Individual sensors can detect their immediate neighbours in the network. Using this neighbour information provided by the nodes in the network, localisation algorithms attempt to recreate the topology, identifying suitable coordinates to be allocated to the individual nodes.

Both Localisation and GPS-based position schemes provide the GHT with a location of the device in the physical space. If the nodes are spread geographically, the storage load will then be distributed evenly. If they are not, the network will suffer from imbalance as individual nodes

will be the closest identity for a large portion of the address space. In an effort to preserve the robustness of the distribution, the scheme proposed should retain the distance assurance properties of the GHT with the balancing properties of a standard DHT.

This chapter details EBL–GHT (Evolve Balance Localise–Geographic Hash Tables)—a mechanism to provide position relative virtual identities that preserve or improve reachability and accordingly provide balanced key allocation for nodes in position relative topologies whilst maintaining device locality. It is intended that, once deployed, the identities would be used by a routing protocol such as GPSR.

6.1.1 Design Goals

The purpose of this work is to design and evaluate a mechanism that will allocate a unique identifier to the individual nodes within the target sensor network. The ID allocated to the node will enable it to take part in a position-relative GHT (using greedy forwarding). The importance of the relationship between the identity and its physical position relates to the ability of the nodes within the network to use a multiple keying function that separates the placement of data within the DHT. Equation 5.4 details a key rotation algorithm, where P nodes can create keys k based on the md5 of the nodes' Universal Resource Identifier, bound by the dimensions of the key space.

As detailed in Chapter 5, if the data is distributed at a number of points across the two dimensions within the DHT and the DHT space relates to the physical space, this will lead to better redundancy given localised failure and no worse redundancy for random failure. It is the intention that the scheme be used both in industrial and home settings, where it would be useful for the state of devices in the network to be preserved in the event that a proportion of the system is lost.

For example, in a home Internet of Things network, an oven could be left on, causing a house fire. The information relating to how the fire started could be passed into the remaining network and, depending on the extent of the damage to the building, the state and spread of the fire could be modelled to provide information relating to the cause, thus facilitating a reduction in the risk of future incidents. Alternatively, the information could be extracted in real-time to provide essential information to first responders. This would require that the responders know the mechanism used to key information and the uri used by the individual devices.

As described in the background research, providing the exact location of a node via GPS is costly and can be difficult to implement indoors. In an effort to overcome the lack of an exact geospatial address, the research contained in this thesis has indicated that we can approximate the relative positioning between nodes given a complete map of the wireless network. Research indicates that the Fruchterman-Reingold algorithm [95] is a good approach to estimating the individual node positions within a network given a set of edges, vertices and weights. The algorithm provides a similar layout to the target physical topology; this has been extensively tested in the literature [85].

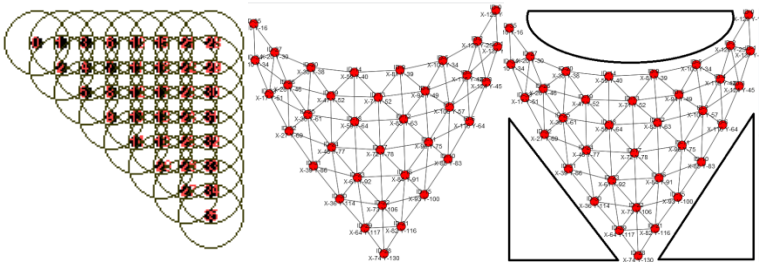


Figure 6.2: Fruchterman-Reingold localisation Example

Figure 6.2 Shows (from left to right) the original node position, the Fruchterman-Reingold-based layout using the nodes and neighbour relations with equal weighting on node links, and an indication of the poor distribution of identities creating spaces in the address space. Figure 6.3 show the placement of keys following using an identity provided by the Fruchterman-Reingold algorithm. The blue lines show the distance between where a key should be located and the node to which the data has routed. As can be seen, half of the nodes in the topology store over half of the keys. If a proportion of those nodes are lost, a disproportionate amount of the distributed data would also be lost.

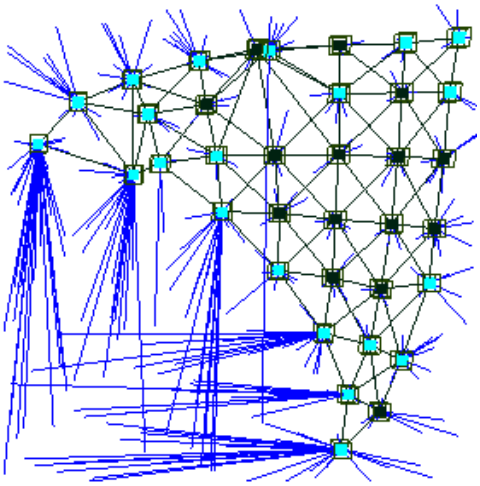


Figure 6.3:Key Distance

Figure 6.2 shows the application of the Fruchterman-Reingold algorithm from the igraph library [107] to a wireless topology. The algorithm provides an estimate of the real-world positioning of the nodes, having been provided with the IDs of all devices and a list of edges connecting nodes. Figure 6.2 also shows the areas that have not had nodes assigned to them; this creates an imbalance in the address space as the data expected to be distributed within these spaces instead maps to nodes that border the vacant space. Such nodes would experience an unfair data load

and would also cause a disproportionate loss of data if any individual device experiencing imbalance fails, as shown in Figure 6.3.

Considering the coordinates for the nodes' initial position, the virtual topology can be tested using simulation to determine the total key space imbalance. This metric provides a measure of fitness that can be used to draw comparisons against attempts to create an improved topology. However, the balance is not the only important measure; rather, the topology must also be evaluated to ensure that data items are reachable by corresponding nodes using greedy forwarding. Importantly, if greedy forwarding fails, routes can still be evaluated using the techniques discussed in GPSR [99].Through reducing the requirement for nodes in the network to perform the calculation required to identify a route when greedy forwarding fails, device longevity can be increased.

Therefore, it can be stated that a good candidate topology is one that that nodes are provided with an address relative to its position in the network. This will maintain the ability to use simple greedy forwarding of data and provide nodes with an even share of the total distributed storage requirement of the network.

To solve this complex problem, the use of Genetic Algorithms will be evaluated to provide a better solution than is provided by the Fruchterman-Reingold algorithm in isolation.

6.1.1.1 Genetic Algorithms Discussion

The search for an optimal state through self-organisation is distinct from any higher system intent or goal. Genetic Algorithms are reliant on the environment to dictate the fitness of individuals to thrive and prosper, given control of the environment random process can be utilised within the fabric of the population and evolve to fit the environmental conditions. Optimisation is a goal that humans set within the environment: whether relating to profit or loss, planning journeys or the organisation of our homes, we optimise to make the best of the environment 'sometimes' reaching the goals set.

At any given moment, environmental conditions can change and our performance is monitored either by us or for us. More often than not, we modify our behaviour in small indistinguishable ways to be more suitable for the environment, and again the fruitfulness of those decisions is as much dictated to us by random chance than by reason.

Global optimisation looks to find the best possible elements within the set of all possibilities evaluated by a set of criteria; this is referred to as the set of objective functions. The objective function evaluates the current genome population of the model. Genomes can then be ordered by their fitness, identifying those candidates that are closer to the optimum. Once ordered by fitness, individual genomes can be selected to reproduce through the application of crossover and mutation. This process continues bound by time, evolution limit, improvement heuristic measure or through the genomes reaching a certain optimisation threshold.

The solution is deemed the best possible, which does not necessarily result in the best solution. Usually, evolutionary algorithms are used when the search space is large, and it would therefore be unfeasible to use an analytical solution or there exists no analytical solution to the problem space. It is the intention of this work to examine the use of genetic algorithms—themselves a type of evolutionary algorithm—with the objective to solve the address space balancing problem seen in position-relative identity spaces where identity is mapped to a coordinate space.

Genetic Algorithms have the capability of utilising multiple objective functions that might be opposing. In the case of Wireless Localisation for distributed storage, these are the distribution of data relative to their intended location and the ability to successfully use greedy forwarding to locate and retrieve data.

6.2 Contribution - Evolve Balance–DHT (EB–DHT)

This section details our novel address localisation scheme. It utilises Genetic Algorithms to generate optimised position relative topologies for use in GHT.

6.2.1 Initialisation

EB–DHT is intended to run alongside existing IoT protocols, e.g. 6lowpan, providing a redundancy mechanism for M2M communications or as an alternative to the client server model found in 6LowPAN. When running alongside Internet Protocol schemes, nodes will utilise the IP address/service identifier tuple to access services distributed within the GHT. Initially, nodes will be identified by a random number that is used for topology construction. Following the pr-DHT construction phase, nodes will be assigned a position-relative two-dimensional GHT address.

In order to start the initialisation procedure, nodes on the network will receive a broadcast from a central coordinating node. For redundancy, there could be multiple coordinating nodes, with individual nodes only responding to a single coordinator. This can be achieved by nodes selecting the coordinating node with the largest ID. The broadcast will be resent by each node in the network where a depth counter will be incremented and forwarded by each node in the network; this will create a distributed tree rooted at the coordinator node. Individual nodes will have generated a random identity and transmitted initial HELLO messages to their connected neighbours, which will facilitate each node in building an immediate neighbour table. Nodes will transmit their neighbour table back to the coordinating node. Upon the coordinating node receiving the neighbour maps from all nodes in the network, it can then reconstruct the entire topology. This completes Stage 1 of the protocol.

In Stage 2, the graph is processed by the Fruchterman-Reingold algorithm to establish the physical node positions. The implementation utilises the Fruchterman-Reingold algorithm from

the igraph library [107] as it provides a good representation of the original network. In Stage 3, the layout is rotated through 360 degrees to find the minimum bounding box. Subsequently, a border is added based on the average distance between neighbours and transposes the layout to the required key size. At this point, the virtual unbalanced topology has been constructed, with each node assigned an address within the key boundary. In an effort to balance the topology, it will be passed into the genetic algorithm where it will be evolved to provide a better balance, with no worse reachability than is provided by the topology at Stage 3.

6.2.2 GA-balancing of Position Relative Topologies

This section will detail the application of Genetic Algorithms to balance the localised topology created in Stage 3. This research has shown that genetic algorithms can be used to find a better solution than the worst case in large search spaces within a bounded search time. Another important aspect of Genetic Algorithms is their ability to fuse multiple metrics to find an optimum that satisfies multiple vectors. The important metrics in this case are the reachability count and the total key space distance imbalance.

The reachability metric *I*, as shown in Equation 6.1,is defined as the difference between the expected total number of keys K_i to be stored by each node and the actual return count R_i from each node in the topology querying each saved state of every other node *n* in the topology.

$$I = \sum_{i=0}^{n} K_i - \sum_{i=0}^{n} R_i$$

Equation 6.1: Reachability Metric

The key space imbalance metric, as detailed in Equation 6.2, is the total displacement *D* of the virtual two-dimensional coordinates of all data items *d(x,y),* currently stored in the DHT at nodes *n(X,Y)* from the virtual node coordinate that each of the data items are being stored.

$$D = \sum_{i=0}^{n} \sum_{j=0}^{d} \sqrt{(X(n)_i - x(d)_j)^2 + (Y(n)_i - y(d)_j)^2}$$

Equation 6.2: Total Key Displacement

It is normal practice that a Genetic Algorithm is seeded using a random population; however, through experimentation, it has been determined that a random population fails to evolve and form a network that meets the fitness and reachability of the initial layout algorithm provided at

the end of Stage 3. The approach taken in this work seeds the Genetic Algorithm with the coordinates provide by Stage 3, which provides the population with a better than random starting point. In Stage 4, the *x* and *y* coordinates of the Stage 3 topology are encoded into the population *P* chromosome of ten candidates. Two chromosomes $C_1 C_2$ are treated as parents and 8 $C_3 .. C_{10}$ as children, applying mutations to each of the 8 in pairs $p_2 .. p_5$ using the following mutation levels $M = Log(nodecount)$ ^2 . The mutations are limited to a random co-ordinate within four radius levels surrounding the existing coordinate in the chromosome; this allows the search space to be gradually expanded. The candidate co-ordinates are then copied into the simulator model. The model initialises enabling the nodes to form a pr-dht. Each node then saves its own state using a three-position replication strategy, utilising the algorithm shown in equation 5.4. Once all save operations have been completed, each node is requested to retrieve every saved state within the system.

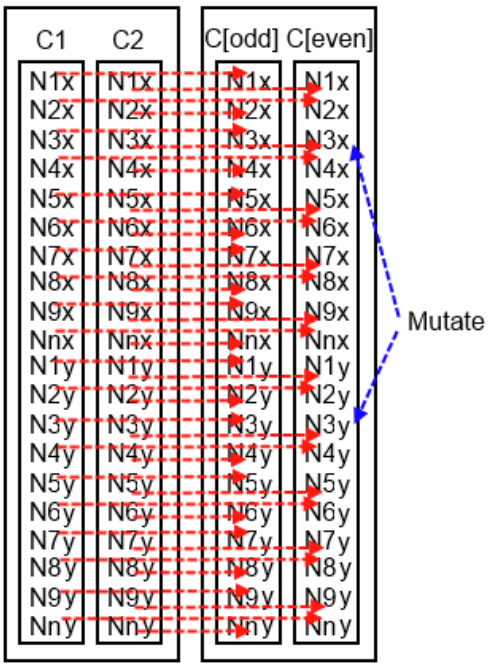


Figure 6.4: Crossover and Mutation

Once this is complete, each genome is scored using the key displacement metric and sorted with the lowest key displacement score being awarded position 1 and the highest being placed in position 10. Any chromosome that has a saved key return metric that is worse than the best will not be processed; this helps to ensure that the reachability of the topology improves or remains constant for each evolution.

Children $C_3 .. C_{10}$ are subsequently populated using genomes $C_1 C_2$, applying an interleaving crossover, as shown in Figure 6.4, that alternates with an even- and odd-node position as the starting chromosome entry. Mutation is applied, and the process evolves for a set number of evolutions. The mutation function generates a random coordinate within a fixed radius of the original coordinate, which reduces the opportunity for the network to get stuck in a local optimum by finding a poor candidate that satisfies an initial low reachability metric and improves

on the key space distance metric. Upon the completion of the mutation phase, the process of testing the topology starts again, which is repeated until the expiration of the time constraints placed upon the process, or until the topology reaches a steady state; that is, no improvement on the score for a set number of evolutions.

Upon the completion of the genetic balancing phase, the new identities are distributed to the devices in the network. In order to reduce the overhead associated with broadcasting each ID, a minimum spanning tree is drawn over the topology and traversed, transmitting the ID of the node and the required neighbour IDs to complete the tree. Neighbours will populate the remaining neighbours using neighbour broadcasts; this enables the protocol to limit the packet size required.

6.3 Evaluation

This section will evaluate the application of Genetic Algorithms to the localisation of nodes within a wireless sensor network. First an evaluation relating to the benefits of seeding the GA with a Fruchterman-Reingold graph layout algorithm are evaluated. Following from this the use of an alternating fitness function to solve the problem of creating balanced position relative identity spaces is then evaluated. Finally, there is a review of the capability of Genetic Algorithms to create balanced position-relative identity spaces for a range of test topologies.

6.3.1 Simulation Environment

The results presented in this chapter have been generated by a Python Discreet Event Distributed system simulator developed for this work. The simulator permits us to simulate the networked environment where nodes are placed physically, corresponding to five different layouts in two different sizes. Nodes operate independently with their own isolated state machines; there is no global knowledge provided to devices. Communication between nodes is achieved through input packet buffers. Where the simulator identifies nodes in the communication range, a buffer relationship is established, thus enabling them to communicate. This work assumes an ideal communication environment with no errors. The simulator is able to drive a genetic algorithm through a series of evolutions where a population of 10 networks is created. These networks then simulate the storage of 9 keys per node, each of which then retrieving the entire key space. In each evolution step, the coordinates of the nodes are mutated according to the method described in the previous section.

6.3.2 Justification for using Fruchterman-Reingold to Seed the GA

It is common to populate a genetic algorithm with a random distribution and then evolve to an optimum; however, in this work, the genetic algorithm is seeded with a Fruchterman-Reingold

graph layout of the topology. The use of graph-drawing to address the problem of localisation for nodes without GPS devices has been shown to be effective in earlier work [86].

	Fruchterman-Reingold	**Random**
Starting Total State Loss	0	74
End Total State Loss	0	44
Starting Individual Key Loss	244	1457
End Individual Key Loss	0	1355
Starting Key Displacement	4282.9732386	4726.809692
End Key Displacement	2291.41074934	4138.55896596
Starting Key Maximum Deviation	4.625	5.125
End Key Maximum Deviation	3.5	4.75

Table 6.1: Comparison between Fruchterman-Reingold and Random placement

In order to validate this decision, the results of an experiment are provided where the GA is seeded using a random topology with 16 nodes. The network is then evolved for 100 cycles, comparing it against the same number of nodes under the same conditions but seeded using the Fruchterman-Reingold layout.

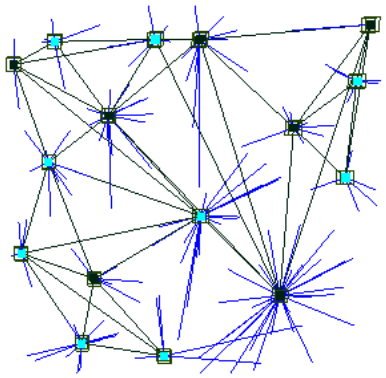


Figure 6.5: Key displacement

The results shown in Table 6.1 show the advantages of using the Fruchterman-Reingold (FR) approach. The initial FR approach has a total state loss of 0; after 100 cycles, the Random topology still has a total loss of 44, having started with a total loss of 74. The individual key loss shows 0 at the end of 100 cycles for the random layout, with 1,355 individual keys for the Random topology. Key displacement and key deviation both show similar properties, being worse in the random initialisations over the FR approach. Such results provide justification for the decision to start with a topology that is localised rather than a random one. Figure 6.5 shows the transformation of the topology, indicating the key displacement to nodes (blue line).

6.3.3 Evaluation of the Alternating Fitness Function

The purpose of EB–DHT is to create an identity scheme that improves the reachability of nodes using greedy forwarding as this will require fewer devices to employ a routing algorithm to circumvent local minimum, whilst at the same time improving the topology balance so that items that are saved into the topology are evenly distributed. The approach taken in this work is centred on using two fitness functions, one to evaluate the total distance that keys are saved from their ideal location and reachability; this is the total number of nodes that are able to see every state from every device on the network.

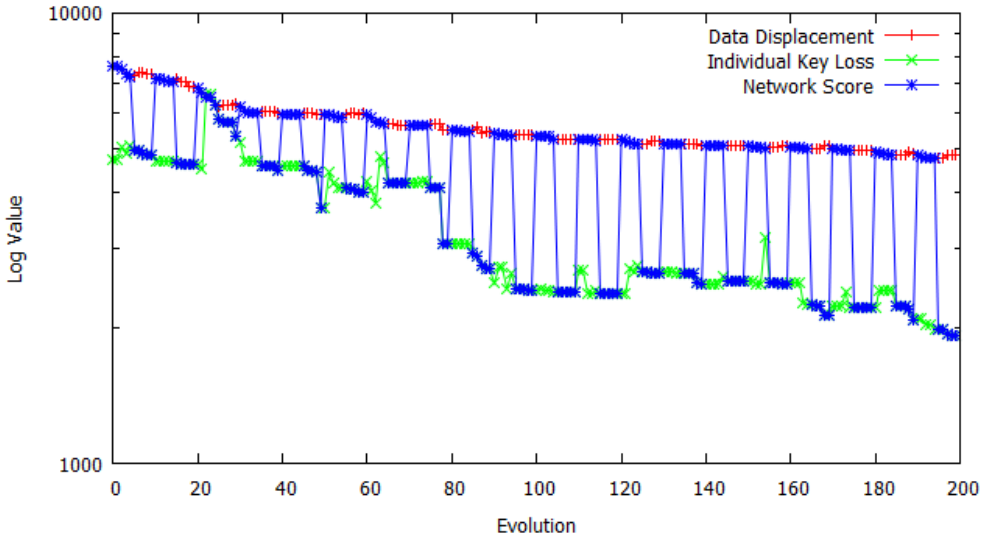


Figure 6.6: Key displacement Fitness Function

Table 6.2 shows the performance of the approach. Alternating key loss and distance provide the greatest reduction in key loss, and also the greatest reduction in key displacement. Accumulating key loss and distance provides the worst key loss difference but also provides good database maximum deviation.

	Alternating Key loss &Distance	**Key loss +Distance**	**Key Loss**	**Key Distance**
Starting Individual Key Loss	244	184	189	314
End Individual Key Loss	0	145	97	215
Key Loss Difference	244	39	92	99
Starting Key Displacement	4282.9732386	3563.1432734	4036.78137885	3968.00602967
End Key Displacement	2291.41074934	2259.0121552	3350.72530145	2341.92788298
Key Displacement difference	1991.56249	1304.131	686.0561	1626.078
Starting Key Maximum Deviation	4.625	4.75	5.0	5.25
End Key Maximum Deviation	3.5	2.875	4.625	4.75
Key Maximum Deviation difference	1.125	1.875	0.375	0.5

Table 6.2: Fitness Function test

Testing key loss alone provides poor results in all tests, whereas testing key distance fails to complete with the alternating approach. Accumulating key loss and distance provides a better deviation but, owing to the poor reachability score, the results show that alternating between key loss and distance provides the better approach. The simulations in this section were carried out using the L-shaped topology pattern, as shown in Appendix C, Figure 1. Each test was carried out for 200 evolutions.

6.3.4 EB–DHT Evaluation

This section will evaluate the capability of EB–DHT in creating position-relative identities for devices in Geographic Hash Tables. The topology generated will fulfil the following requirements.

- Reachability: Reduce the total number of states that devices in the network are unable to retrieve. To provide an assessment of reachability, each node saves a set number of keys into the topology. Following the completion of the save operation, each node on the network retrieves each of the save states from every other device in the topology. Upon receipt, each node keeps track of the number of returned states. Following the completion of the save and retrieval steps, the number of keys that have been distributed and the number that should have been returned are calculated. The network score is based on the number of missing keys.
- Key Distance: Reduce the total distance that all keys in the network rest from their intended destination in the key space. Initially, nodes are provided with an identity that is relative to their position in the network. Data will then route to a point that is closer to the target, eventually reaching a node that is closest (best case) or hitting local minimum and saving on a device that is not the closest to the destination coordinate (worse case). If a node has a disproportionate quantity of the address space, more keys will be placed on this node. Some keys will have been intended for the location that the node occupies, whereas other data would have been destined for coordinates that are distant from the device. The probability for a network to have zero key space distance is extremely low. It would require an infinite number of devices spread out equally across a square area or the data items keying exactly to the address of nodes in the topology evenly.
- Maximum Database deviation: Reduce the storage imbalance in the network. This is achieved by calculating the mean database size and then calculating the maximum deviate. We look to reduce the maximum deviation to distribute data more evenly across the network.

To test the robustness of the approach, five different topology types have been created with two size variations for each topology. These are shown in the figures provided in Appendix C. Each topology is evolved for a total of 200 cycles.

The topology types have been chosen due to their range of local minimum inducing features. A square grid pattern is utilised to give ideal results as this topology provides the Fruchterman-

Reingold algorithm with the opportunity of producing a topology with excellent characteristics. This will serve as both a benchmark and a mechanism to validate the simulator. The square topology will provide a measure for the other topology types, providing exemplar results for reachability, balance and key distance.

The small topology patterns shown in Appendix C, Figure 1, have limited node counts and operate within a smaller network boundary of 100 x 100metres. This limits their opportunity to improve storage balance; however, the results of our experiments show that all topologies improve their reachability, with the 15-nodetriangle topology achieving full reachability. The L-shape topology with 16 nodes, the H-shape topology network with 19 nodes and the Hole topology with 21 nodes are all missing a number of states. The topologies with missing states are the ones most likely to exhibit local minimum. Notably, however, they show excellent improvements over the use of Fruchterman-Reingold alone, with an improvement in the reachability of 559 states for the H topology, 276 for the Hole-shaped topology and 213 for the L-shaped topology. This would reduce the number of route calculations required to identify the alternative paths to establishing those data elements not accessible by greedy forwarding alone. The final key deviation for the square topology with 25 nodes is worse that the initial topology. However, at evolution 118, the results show a maximum deviation of 1.3 with no missing keys. The Hole topology with 21 nodes also has a worse maximum deviation as with the square topology a previous topology had a better deviation with a score of 1.8 with an equivalent missing key count. This leads to the possibility that saving historical best deviation and missing key count and comparing that to the final evolved topology would be the best approach.

	Square 25 Nodes	**L Shape 16 Nodes**	**H Shape 19 Nodes**	**Triangle 15 Nodes**	**Hole 21 Nodes**
Starting Individual Key Loss	0	244	562	132	458
End Individual Key Loss	0	31	3	0	182
Key Loss Difference	0	213	559	132	276
Starting Key Displacement	1759.881	4282.973	2338.070	2323.243	2384.596
End Key Displacement	1508.722	2291.410	1787.968	1367.929	1906.262
Key Displacement difference	251.159	1991.563	550.102	955.313	478.333
Starting Key Maximum Deviation	1.84	4.625	3.263	5.866	2.476
End Key Maximum Deviation	2.48	3.5	2.421	3.2	3.047
Key Maximum Deviation difference	-0.64	1.125	0.842	2.666	-0.571

Table 6.3: Small topology GA Test

The results shown in Table 6.4 show the effect of this approach on larger topologies bound by an area of 160 x 160 metres. As was seen in the small topology, the square with 64 nodes provides excellent reachability and storage balance. The square topology provides a reachability loss count of zero before and after the 200 evolution steps. The topology also exhibits a slight improvement in storage balance. The L-shaped topology with 48 nodes shows an improvement but an increase in key balance.

	Square 64 Nodes	**L Shape 48 Nodes**	**H Shape 54 Nodes**	**Triangle 36 Nodes**	**Hole 52 Nodes**
Starting Individual Key Loss	0	1972	4737	2626	5760
End Individual Key Loss	0	1397	1921	576	3895
Key Loss Difference	0	575	2816	2050	1865
Starting Key Displacement	4118.317	7482.336	7602.327	6683.651	8679.543
End Key Displacement	3756.372	5042.171	4850.933	4144.557	6455.108
Key Displacement difference	361.945	2440.17	2751.39	2539.09	2224.44
Starting Key Maximum Deviation	3.0	4.875	5.074	6.388	4.653
End Key Maximum Deviation	2.75	5.291	4.481	4.222	3.730
Key Maximum Deviation difference	0.25	-0.416	0.593	2.166	0.923

Table 6.4: Large Topology GA Test

As was observed through the small topologies, a previous evolution had improved balance characteristics, achieving a balance of 4.4. However, this time, the key loss is slightly higher with a loss of 14,000 keys at evolution 107. There is also a deviation of 4.05 with a key loss of 2,000, which leads to decisions as to the importance of reducing key balance against the cost of key reachability. The H-shaped topology with 54 nodes, the Triangle topology with 36 nodes and the Hole topology with 52 nodes all improve their key counts. The H, Triangle and Hole topology also improve their key imbalance.

6.4 Conclusion

This chapter detailed a novel technique for generating balanced position-relative identities for use in Geographical Hash Tables in wireless networks. The approach, EB–DHT, utilises a novel alternating fitness function combining the benefits of two metrics to improve an initial topology. The approach also introduces the novel technique of evolving based on the output of wireless network simulation to generate topology improvements.

The results, through simulation, have shown that the approach detailed in this chapter provides and improves on the initial Fruchterman-Reingold layout for all layout types, both for key loss and database deviation. This will reduce the total power consumed by the network when making greedy forwarding decisions, reducing the requirement of the angle calculations needed by local minimum avoidance techniques. This approach also reduces the database deviation, resulting in a fairer distribution of state amongst individual devices.

Through experimentation, the following conclusions have been reached:

- The Fruchterman-Reingold graph layout algorithm used in isolation produces a good representation of a target topology given the neighbour relationships of all devices. However, the topology provided poor balance and reachability when using rendezvous communications on top of a Geographical Routing Protocol using Greedy Forwarding.
- Utilising Genetic Algorithms, it is possible to modify the initial coordinates provided by the Fruchterman-Reingold algorithm so as to improve balance and reachability when using rendezvous communications on top of a Geographical Routing.
- Alternating fitness functions can improve the ability of the GA to find an optimal solution when compared with the use of individual metrics or through the combination of metrics.

This chapter provides a balanced identity space to support initial topological state for a wireless network. This thesis is concerned with the continued operation and preservation of state following the destruction of part of the system. It is therefore essential that the initial topology can compensate for disturbance.

The next chapter will consider the problem of the localised failure of nodes in Geographical Hash Tables and will detail a mechanism for mitigating their impact.

Chapter 7

Correcting GHT Imbalance Through Topological Isomorphism

In Chapter 5 and Chapter 6, the use of a keying mechanism to distribute a rendezvous device state within a cross-layer routing and storage Hash Space was detailed. One of the main problems identified in Chapter 5 relates to the balancing of the address space following the failure of nodes within the network. This chapter will describe a distributed mechanism to resolve the imbalance that can occur in unbalanced DHT once the network is initially balanced during the identity allocation scheme proposed in Chapter 6.

An essential requirement of the IoT (Internet of Things) is the reliable M2M (machine-to-machine) communication of a device's state in the event of system failure. We have established in Chapter 5 that decentralised rendezvous P2P communication mechanism, constructed using Distributed Hash Tables (DHT), provides acceptable levels of redundancy given partial network failure. In an effort to accommodate the resource-constrained nature of both the device and communication channel commonly deployed to the Internet of Things, in Chapter 5 it is suggested that there be the use of a position-relative DHT, also referred to as Geographic Hash Tables (GHT), as an alternative to using overlay DHT in wireless IoT networks. The position-relative DHT uses local knowledge to forward information closer to the target, which is commonly referred to as 'greedy forwarding'[108], as discussed earlier. When using position-relative DHT, individual device IDs are relative to a node's position in the topology, with such information shared with 1-hop neighbours in the topology. This permits a node to receive a data packet to either forward the information to a node that is closer to the destination or to save the information in its own local database if there is no neighbour with a closer address.

In the ideal case, nodes would be allocated address using the approach shown in Equation 7.1.

$$N(x,y) \in f((x,y)): \sum_{i=1}^{X} \sum_{j=1}^{Y} x = i, y = i, i = i + a, j = j + a$$

Equation 7.1: Ideal Address Allocation

With x and y being the bounds of the topology and defining the interval between v in N, providing identities in this way has the added advantage that logical separation of the DHT is reflected in its physical position, which is not the case with overlay DHT, where identity is different to physical location. The ideal topology is used to define an initial state; however, any failure within the topology would disrupt its organisation, causing reachability and balancing concerns. Reachability concerns relate to the ability of communicating parties to route data between themselves either directly or via a rendezvous point. Balancing concerns relate to the quantity of data that must be held by individual nodes and the amount of routing that needs to be undertaken. Notably, both activities consume valuable resources. The greedy forwarding technique will cause reachability and balancing problems owing to the local minimum phenomenon [108].

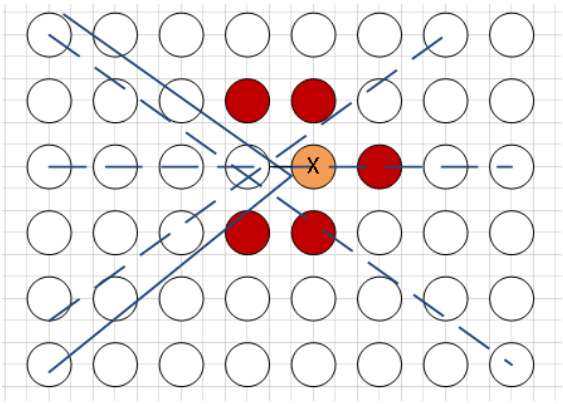


Figure 7.1: GHT Local Minimum

Figure 7.1 shows the number of failed nodes (shaded circles). On the left-hand side of Figure 7.1, nodes are contained within the cone that could map their state to destination nodes contained within the dashed cone on the right hand side. The topology shown in Figure 7.1 would result in packets distributed by nodes in the left-hand cone destined for positions in the right-hand cone being saved at the node marked x. This results in the node marked x having a disproportion share of the data. Most nodes that border the failure will have an increased

number of states held locally than those that lay one hop away from the fault. In order to resolve this imbalance, the topology must be capable of self-organising so as to minimise the effects of local minimum and to reduce the distance a key will rest in relation to its intended destination.

The distance that a key resides away from its intended location will have an effect on its reachability. For example, if corresponding nodes are on the same side of a fault in the topology, both put and get requests will incorrectly map to the same node, which maintains reachability at the cost of balance. However, if the corresponding nodes are at opposing angles to the fault, one will map to the correct location and the other will incorrectly map to a node on the fault line. The use of multiple keys that are geographically separate on the network will help maintain reachability in the face of network failure; however, it would be in the best interest of the network to compensate for the loss of nodes to both correct the imbalance that would be caused by both put and get requests, and the storage-routing requirements placed on individual devices that border the fault.

This chapter describes a novel approach to the DHT-balancing problem by identifying a mechanism to correct the fault through Topological Isomorphism. This chapter describes the use of topological isomorphism to resolve the balancing issues caused by node failure in position-relative DHT that use simple forwarding rules. This chapter describes a protocol that applies topological isomorphism to routing to wireless IoT sensor networks. This chapter concludes with an evaluation of the approach through simulation and a summary of the work.

This chapter contributes a routing protocol that mitigates the effects of failure with Geographical Hash Tables through the application of Topological Isomorphism to the address space of those nodes surrounding the failure.

7.1 Objective

The objective of this chapter is to extend the work identified in Geographical Routing within wireless sensor networks. The literature review identified current geographical routing protocols are not suited for data storage, as they do not modify the identity space to counteract voids. The review identified that existing schemes do look to reduce the effect of routing around voids, but these schemes can be fairly rudimentary, using round robin approaches or local knowledge of neighbour work load. In this chapter we will look to extend the work of curveball routing to alter the areas around the fault to distribute routing and storage overhead.

The review of related work in chapter 3 identified the following issues:

- Current Geographical routing schemes look to reduce the overhead of routing and do not concentrate on storage.
- Routing is either reduced by global knowledge e.g. new keying mechanism shared by all sensors or round robin approaches. This leads to communication overhead and the network not being able to respond to small pockets of failure.

7.2 Contribution - DHT Load-balancing using Topological Isomorphism

7.2.1 Overview

Topological Isomorphism is defined as a continuous function between topological spaces. This section will detail a Topological Isomorphic function that can be deployed between nodes in a distributed system to transform the distributed Identity space to correct for storage and routing imbalance. The initial topological space for the problem domain is defined as follows:

Given a set of operational nodes $G = (v)$ with a subset of nodes $F = (v)$ that have an effect of radius *R* on a set of nodes $A = (v)$.

This provides a set of nodes that will accommodate for the failure of *F* in *G*, AS shown equation 7.2:

$$C = \{(F - G) \cup (G - A)\}$$

Equation 7.2: TIF Accommodating Nodes

Equation 7.2 details a function to transform *C* to aide in the balancing and routing of data to mitigate the impact of the failure with regard to the centre of the failure. The centre coordinate of the centre of the disturbance is denoted as$d_{x,y}$ with radius *r* in respect to *C* and *k* as the radius of the fault preserving area.

$$f: C \to V$$

$$N = Size(C)$$

$$C = \{(x, y): x, y \in R\}$$

$$f\big((x, y)\big) = \begin{pmatrix} d_x + \left(\frac{\frac{k}{4}}{k + \frac{k}{4}}\right) r + k\left(\frac{x - d_x}{r}\right), \\ d_y + \left(\frac{\frac{k}{4}}{k + \frac{k}{4}}\right) r + k\left(\frac{y - d_y}{r}\right) \end{pmatrix}$$

Equation 7.3: Topological Isomorphic Function

Equation 7.3 permits individual nodes to transform the identity of a data destination coordinate from the default topological space to one that will distribute data from a fault region to a compensating region. Importantly, the equation requires only that individual nodes have knowledge of the central point of failure and its radius to calculate the required transformation. The ability of nodes to acquire this information is essential in terms of computing the new topological coordinate. This capability would be implementation-dependent. In the case of wireless sensor network deployment, this detail will be provided in the next section.

Figures 7.2 and 7.3 show the application of the function to a set of ordered and unordered node identities.

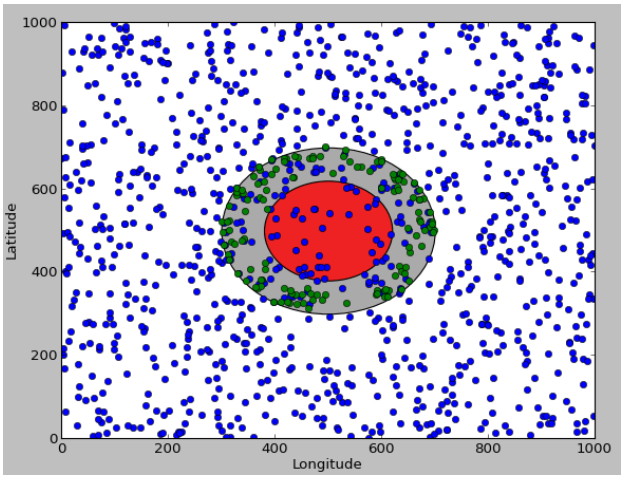


Figure 7.2: Local Minimum random Distribution

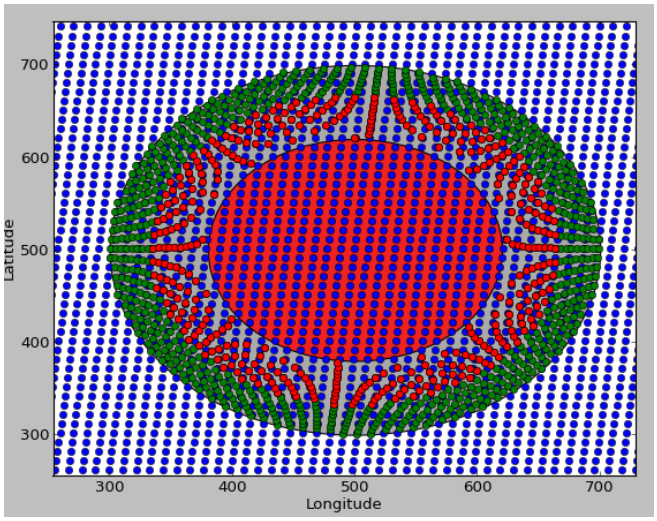


Figure 7.3: Local Minimum Distribution

7.2.2 Routing Post-topology Transformation

Individual nodes will still be required to identify the next hop in the topology to which packets should be forwarded. Data that maps inside the region $f + C$ will be relocated to the area responsible for the failure C. The results of this transformation are shown in Figure 7.2 and 7.3 for both structured and unstructured topologies. Using this new address, data can be mapped around the failed region. In order to enable data to route to this transformed address or to route beyond the fault avoiding local minimum angle, a routing rule is provided, which is accomplished using the central coordinate of the failure, enabling a node to calculate the best trajectory for the data item to take around the fault.

If the data item destination is located outside of the fault, which can be determined using the central point of the failure and its radius, the original address is retained. If the data point is destined for the fault, the Topological Isomorphic Function is used to generate the new identity for the data element. Subsequently, the angle between the destination address and the address of each node neighbour is identified, forwarding the data to the neighbour with the smallest angle. This process repeats until the destination angle is achieved.

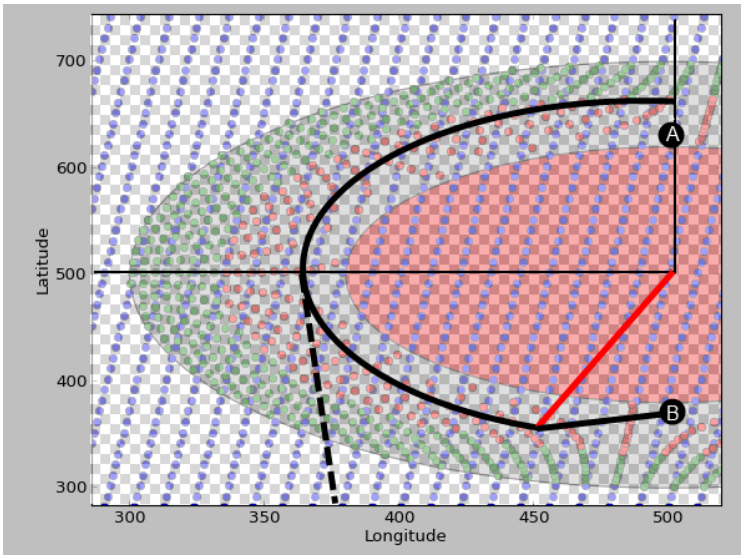


Figure 7.4: Topological Isomorphic Routing

Once the data is at a node with an angle > than 90 degrees the fault will pose no threat of forming a local minimum. At that point, the routing process reverts to greedy forwarding. Data destined inside the lens will need to rotate to a smaller angle in respect to the target location in order to avoid local minimum. In Figure 7.4, this is shown by the data item located at Point A routing to within an angle < 20 degrees of Point B. Data can either collapse or rise in respect to the fault centre using greedy forwarding, as shown in Figure 7.4.

The Topological Isomorphic functions defined in this section insure that there is an active node mapped to every address in the DHT space. Identifying the shortest angle path with respect to the failure will provide a route around the failure, avoiding local minimum.

The following section will detail a Topological Isomorphic Routing Protocol (TIR) that implements the functions described in this section.

7.3 Topological Isomorphic Routing (TIR) Protocol

This section will describe the implementation of the Topological Isomorphic function detailed in the previous section. The TIR protocol aims to address the balancing and routing issues present in current greedy routing protocols where holes form in the topology following an initial address-allocation phase. The protocol aims to create TIR-corrective regions to help balance the address space, enable routing and reducing the routing overhead around the area of the failure.

This work is not concerned with the mechanism used to allocate individual addresses to nodes within the topology. This can be achieved with the use of GPS Sensors, localisation methods or manual allocation. The protocol will support the basic DHT primitives get and put. For the purpose of simulation, the distributed rendezvous protocol described in Chapter 6 will be used. The protocol utilises a DHT to enable a node to distribute a rendezvous state to multiple locations in the topology. For the purpose of this simulation, each node will copy its state to three points in the topology. The following sections detail the operation of the protocol.

7.3.1 Initialisation and Operation Pre-failure

TIR uses greedy forwarding, meaning it needs only to know the identity (coordinate tuple) of its neighbouring nodes to make a forwarding decision. Nodes periodically broadcast HELLO packets that include the sending ID, the coordinate of the node, neighbour loss state (the number of neighbours that have not sent HELLO message within time boundary), the size of the neighbour's database (the number of data items that have been stored at that node) and the TIR region index (centre coordinate of the disturbance). The Application space will call put and get functions on TIR so as to enable higher level functions, such as the distributed rendezvous protocol described earlier. Upon calling the put function, TIR will evaluate its neighbours and its own identity, establishing which is closest to the destination. If a node determines that it is the closest to the destination, the data item will be stored locally; if, on the other hand, it is determined that a neighbour is closer to the destination and the node is not in a TIR region, the data item will then be forwarded in a save packet to the selected neighbour. The Save packet comprises the coordinate destination tuple and the data item. The operation of TIR will change on the detection of a hole in the network.

7.3.2 Hole Detection

The first step in accommodating a failure is the identification of the failure and its dimensions. In an effort to enable the deployment of the topological isomorphic function described in section 7.2, it is essential that the central point of the disturbance and its radius are captured. The greedy forwarding local minimum problem, combined with local failure of neighbours, provides an opportunity to identify those neighbours on the fault. Nodes do not need to experience imbalance to classify them as being on the fault border. If they experience the loss of a neighbour and through neighbour updates identify that surrounding nodes are experiencing an imbalance, they, too, can transition to the border state. Figure 7.5 shows the imbalance those nodes on the edge of a failure experience.

The blue lines detailed in Figure 7.5 show the distance that a key lies from its intended destination coordinate. The values on the diagrams on the right indicate the number of items that have been saved on each node. The disabled nodes show the value 3 as they are the only node in their respective networks so save the value locally. The nodes on the edge of the disturbance all carry more keys than those in the surrounding area.

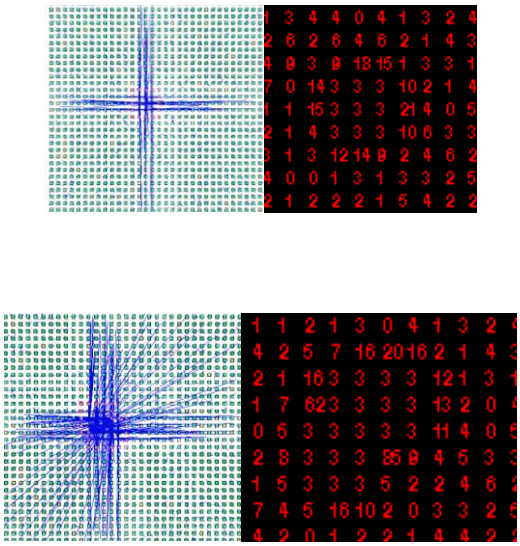


Figure 7.5: Key Imbalance

Once a node transitions to the fault border state, it calculates the extent of the fault key range; it does this by expanding the fault range experienced by its neighbours. The extent of the fault key range is expressed as the tuple pair bottom-left and top-right coordinates. Fault border state nodes exchange fault border packets; they identify the total region of the fault by expanding their perception of the area using the received tuples from other fault border nodes. These are accumulated and retransmitted. Figure 7.6 shows the distributed process of fault identification.

Nodes use a hold-down timer that is reset when a node receives a coordinate tuple that expands the fault region beyond the current perception.

The process finishes when the hold-down timers expire and all nodes on the fault border have the same extent tuple that defines the area of the fault.

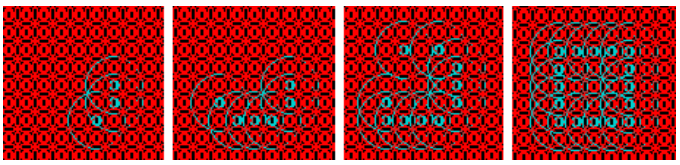


Figure 7.6: Fault Border Detection

7.3.3 Establishing TIR Corrective Region

In order to establish the TIR region, nodes that will form the region will need to be identified and provided with the values required to perform the address space transformation. Individual border nodes broadcast TIR packets that contain the central coordinate of the fault and its radius. Nodes will have a preconfigured ratio that will be applied to extend the radius of the fault to the TIR region. Nodes receive the TIR broadcast and determine whether their identity falls within the TIR region: if it does, the node retransmits the TIR broadcast. The broadcast self-terminates as it reaches the radius of the TIR region. Figure 7.7 shows nodes in blue that have self-organised to form the TIR region for the fault.

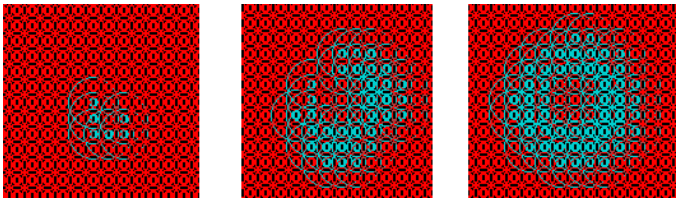


Figure 7.7: Redundant Rendezvous

Nodes independently identify the border of the fault and subsequently initiate the distributed formation of the TIR region. Once in the TIR region, individual nodes then can apply the Topological Isomorphic function detailed in Section 2 of this chapter.

7.3.4 Routing

Nodes outside of the TIR region communicate through the utilisation of the greedy forwarding mechanism as described in Section 4b. When a save packet enters the TIR region, the process

shown in Figure 7.8 is initiated. Primarily, a node needs to determine whether it is inside the TIR region; if this is the case, the data destination address then will be modified in accordance with the function outlined in Section 2, which balances the address space of the TIR region to a range outside of the fault but within the TIR range. This is achieved due to the TIR function shifting the address space of those nodes that have failed to an area outside of the failed region. This permits a subset of TIR nodes to take responsibility for the failed nodes and not place the burden on those nodes that are positioned directly on the fault border.

Following the application of the TIR function, the address of a data item will map to the identity of an active node within the TIR region. If the node falls inside the TIR region and the data item maps to an area outside this TIR region, the data destination address is not modified.

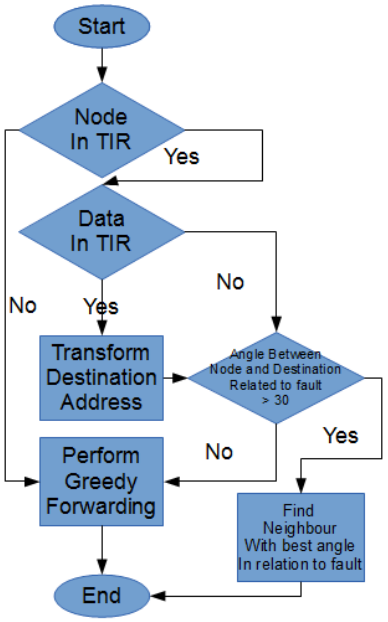


Figure 7.8: TIR Flow Chart

In an effort to prevent the local minimum problem, a node checks to see whether the data item is within 30 degrees of its own location in respect to the centre of the fault. This results in a data item rotating around the fault by the shortest angle until it can rest within the TIR region or exit using the normal greedy forwarding algorithm. This action has an additional benefit of distributing the routing function between nodes. Normally, nodes located on a fault become the shortest path, and all routing occurs on this line. This scheme evaluates a neighbour's angle to its destination, resulting in data routing via the shortest arc around the fault. This would prevent data from falling against the fault when routing to endpoints hidden by the fault.

7.4 Evaluation

The following section provides an analysis of the results following the experimentation of Topological Isomorphic Routing applied to wireless sensor networks. Experiments were carried out with the use of an idealistic network of 1,024 nodes, arranged into a grid formation with equidistant spacing. Nodes are separated by a distance of 12 meters, and all nodes are contained within a 400x400 metre region. Each non-edge node has a set of 8 neighbours. Simulating in this way provides a repeatable test mechanism.

The purpose of these experiments is to evaluate the use of Topological Isomorphism to correct failures in position-relative DHT. TIR will be evaluated against the GPSR protocol[99]. This is one of the most popular geographical routing protocols. TIR is also evaluated against the worst case scenario of a network without a fault mitigation protocol, denoted as NR in figures and in the remainder of this text. Each protocol was tested using the six fault patterns detailed in Figure 7.9. They provide faults with the follow node counts: 5, 9, 17, 21, 29 and 36.

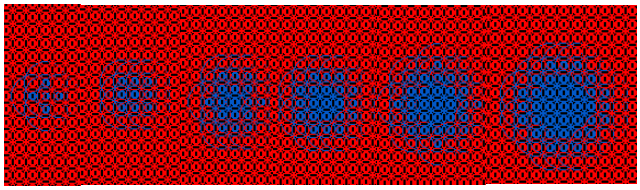


Figure 7.9: Fault Patterns

The fault types introduce various local minimum problems. Local minimum in geographic routing is the perception of a node that its coordinates are the closest to the destination. When in fact taking a number of less optimum steps would result in a destination that is more optimum than that of the local minimum.

7.4.1 Analysis of Routing Capability

We will first analyse the capability of TIR to route around a failure that has formed post topology creation. This will test the capability of TIR to avoid local minimum. We will compare TIR to the GPSR protocol as GPSR is a popular geographical routing protocol that is used extensively in the literature to validate new approaches.

We find that TIR and GPSR have equal total routing path overhead (figure 7.14) until the larger disturbance size of 6 is reached. At this point, TIR exhibits a 4% efficiency in routing over GPRS; this is owing to the increasing quantity of nodes lining the perimeter of the failure and the additional overhead GPSR exhibits when finding the appropriate node that is closer to a data address contained within the failure. GPSR will attempt to route around the fault to find an optimum identity. TIR improves on GPSR by routing around the failure in the appropriate direction, as well as mapping data to nodes that are located further from the fault border. This

has a twofold benefit, data is distributed away from the fault border, with data mapping inside the fault resting at multiple nodes instead of it resting at a single fault border node and the reduction on the hop count of all data items destined for the fault border and fault region. Mapping data to nodes away from the fault border reduces the total hop count incurred by the system.

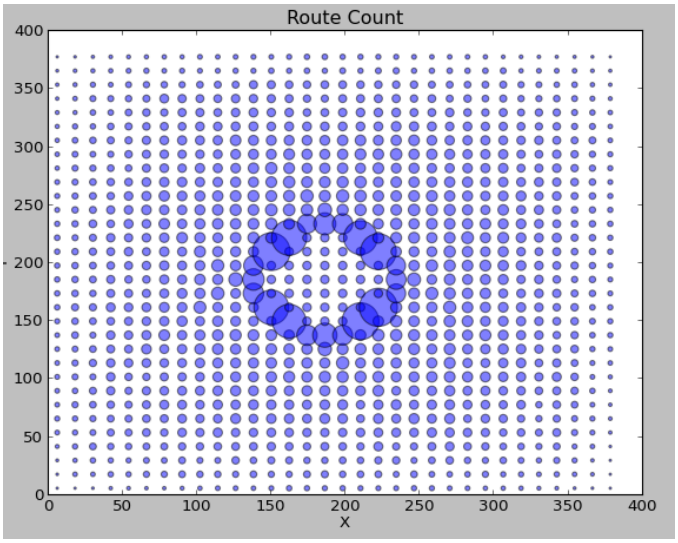


Figure 7.10: Data-forwarding Overhead GPSR

Data-forwarding overhead gives a better view for understanding the impact of a fault on the surrounding nodes. Figure 7.10 shows the impact of using the right-hand rule in GPRS for making forwarding decisions. The diagram shows the relative quantity of forwarding actions performed by nodes, the centre of a circle indicates a nodes position, the size of the circle indicates the relative forwarding actions performed by the node. As would be expected those nodes on the periphery have fewer forwarding actions to perform, with the number of forwarding actions increasing toward the centre of the topology.

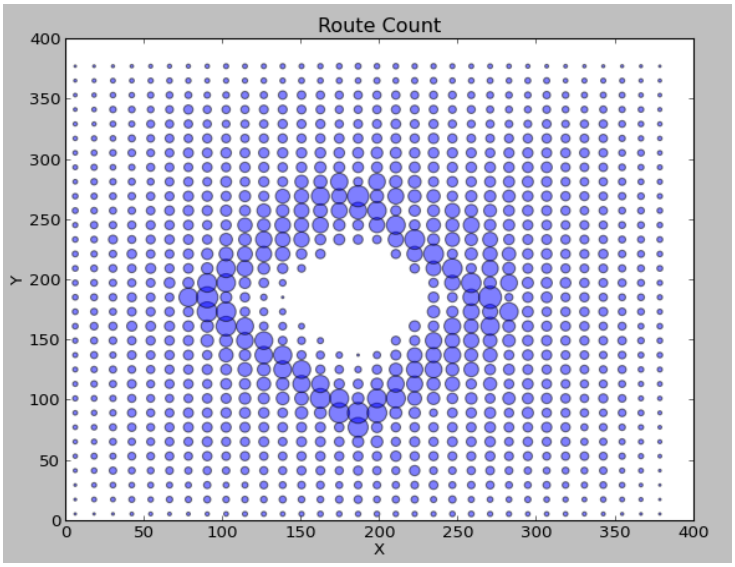


Figure 7.11: Data-forwarding Overhead TIR

The size of the circles for nodes positioned on the edge of the failure in Figure 7.10 clearly shows the imbalance between nodes on the fault and their neighbours located one hop away from the fault. This would have a negative effect on the battery life of these devices resulting in an unfair distribution of storage load. The possibility of further failure would also have serious implications for the system as a larger proportion of data items would be lost with subsequent fault expansion. Due to the effect of the distribution function shown in equation 7.3, TIR balances the forwarding of data to nodes with the TIR region, which has a drastic impact on the forwarding distribution shown in Figure 7.11. Nodes surrounding the failure have a better balance of forwarding responsibility.

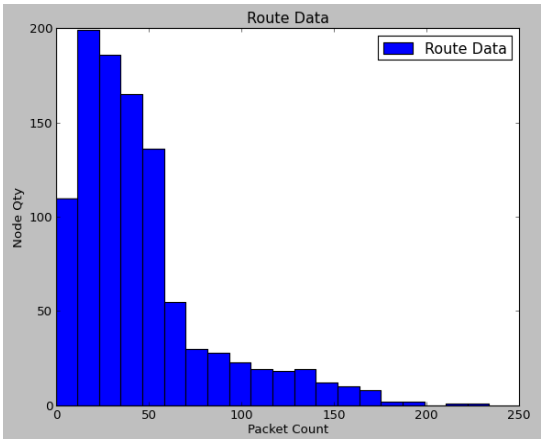


Figure 7.12: Data-forwarding TIK

Figure 7.13 shows the number of packets that nodes forward in GPRS. Those nodes that rest on the border can be clearly identified; there are 8 in total, and they forward between 650 and 850 packets each. This would have a negative impact on those 8 nodes resulting in them depleting

internal storage, and consuming an unfair amount of energy when receiving and forwarding data. Figure 7.12 shows TIR as having a better distribution with nodes having no more than 250 forwarding decisions.

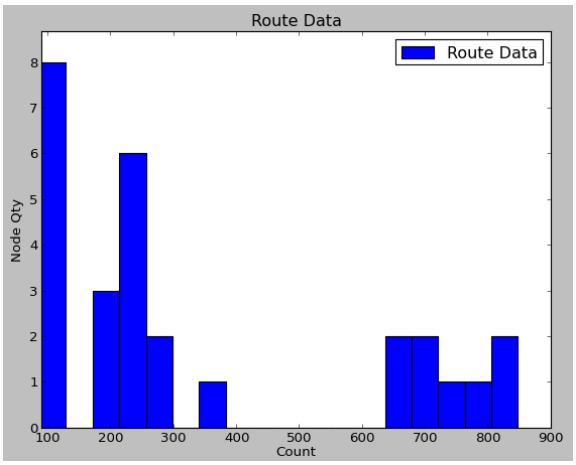


Figure 7.13: Data-forwarding GPSR

An additional benefit of the TIR function is the reduction in the routing responsibility of those nodes surrounding the fault. This characteristic is the opposite to operation of GPSR, where traffic routes against the fault. This would enable nodes adjacent to the fault greater opportunity to distribute their own state into the topology. In the case of continued non distributed failure, providing these nodes with a greater opportunity to transmit will provide the best opportunity to save the state of the devices most immediately under threat.

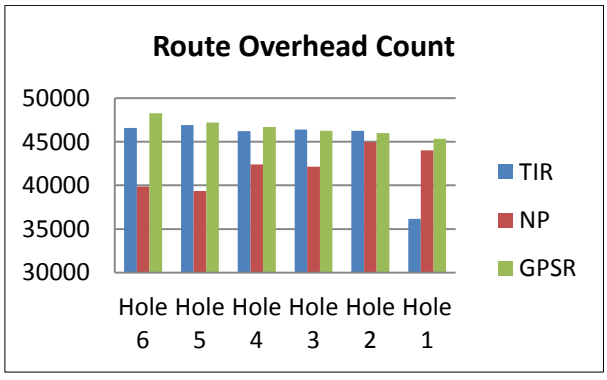


Figure 7.14: Data-forwarding Overhead

7.4.2 Analysis of Data Distribution

As shown in figures 7.17 and 7.18, data distribution between TIR and GPRS are similar; however as a result of the Topological Isomorphic Function detailed in equation 7.3, TIR includes nodes with slightly higher databases towards the outside edge of the TIR region. This is a result of data items mapping away from the disturbance, which reduces the routing costs but results in nodes closer to the disturbance having databases in the Region 0 to 9, and those toward the furthest extent of the TIR region distributing the offset data, resulting in nodes having between 0 and 12 data items. This reduces the risk of data items being lost to nodes that are adjacent to the failure. The additional storage requirements would be dependent on the amount of data being stored. Our tests concluded that a Real Time OS would consume ~30kb of memory if we assume an average microcontroller with 128kb of FLASH memory. Then the device would have the capability to store ~98kb of data. Assuming that devices would monitor four ten bit analogue to digital conversions and on average 8 bits of IO. Then devices would be adequately capable of storing the additional state information.

Figure 7.15 shows an output from the simulator identifying individual device state and the storage vector of data that has been remapped from within the fault. The diagram shows the physical location of nodes, the colour code is as follows; black nodes in the centre of the diagram have failed, red nodes that surround the back nodes have identified themselves as border nodes. Green nodes show the position of the nodes that have identified themselves as being within the TIR corrective region. The Blue nodes are the displaced identities of those nodes in the TIR corrective region. The Blue Lines shows the displacement of data that has been mapped from the failed region to the TIR corrective region. The placement shows excellent symmetry between the intended location pre-failure and the compensated location post-failure.

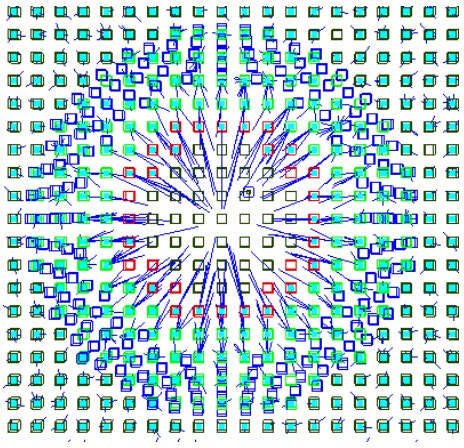


Figure 7.15: Key Displacement TIR Hole 6

Figure 7.15 shows that TIR border nodes have fewer data items mapping to them than their one hop neighbours within the TIR corrective region. This is also shown in figure 7.16 where the size of the node indicates the relative storage amount. Figure 7.16 shows an empty circular region

surrounding the failed nodes. The nodes inside the clear region are nodes in the failed zone, they are still active in the simulator although disconnected from the topology, and as they are running an instance of TIR they save their own data items locally (as they own the entire key space). The clear region is populated with border nodes that have not been requested to store data.

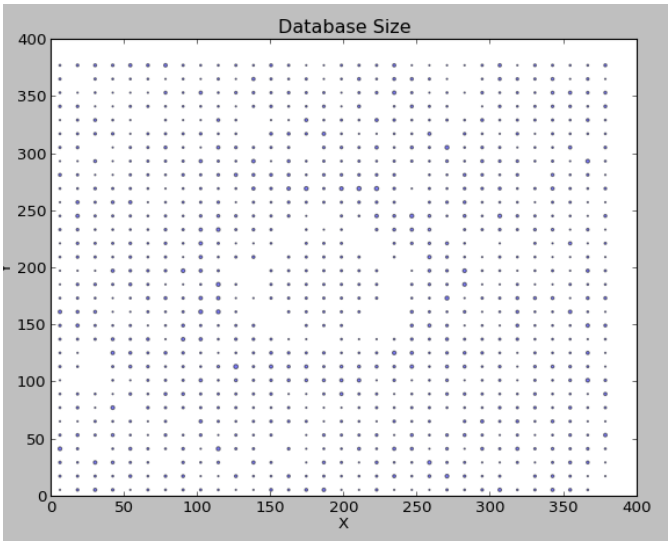


Figure 7.16: Data Distribution TIR Hole 6

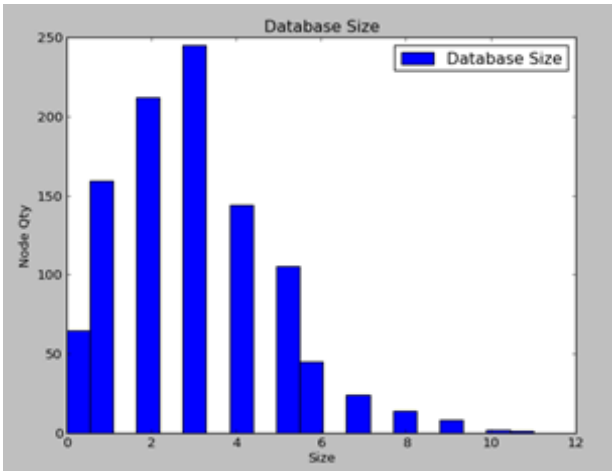


Figure 7.17: Data Node Distribution TIR Hole 6

Due the nature of the Topological Isomorphic function and the even distribution of keys generated by the hash function, nodes close to the inside edge of the TIR corrective region have few or no keys allocated to them. This could be corrected using a function that is proportional to the distance from the centre of the failure; however, this would have a negative impact on routing overhead. The usefulness of this phenomenon will be expanded in future work.

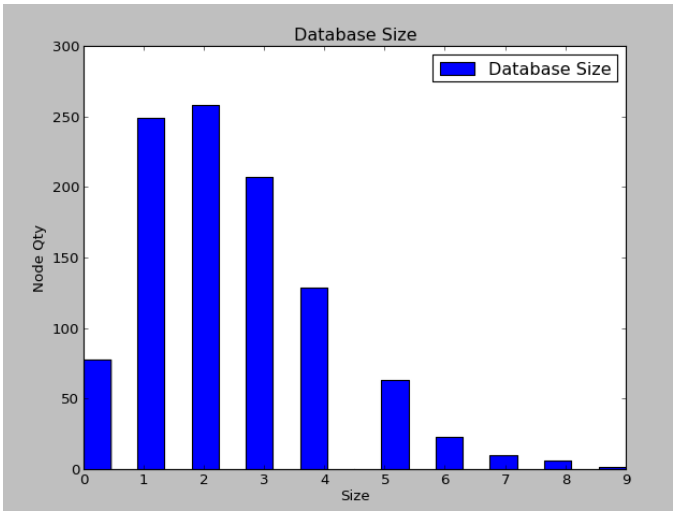


Figure 7.18: Data Node Distribution GPSR Hole 6

7.4.3 Analysis of the Implementation

The TIR protocol provides an implementation of the topological Isomorphic function described in Section 3. To initiate the protocol, nodes must communicate the loss of neighbours and the extent of the nodes responsible key region. This information must be propagated to other nodes on the perimeter of the failure. Once nodes have identified the failure, a node will need to read in at least the border node count/2 border packets so as to enable the extent of the border to be discovered. This packet must contain two sets of coordinates, namely the TIR region ID and the packet ID, which results in a packet size of 48 bytes given a 320-bit coordinate tuple. Nodes will then transmit a TIR region packet that includes the coordinates of the centre of the fault, fault radius, TIR region ID and a packet ID, resulting in a packet size of 64 bytes. Even if the size of the coordinate tuple is doubled, the maximum packet size required by TIR is 112 bytes. This is well within the 127-byte maximum payload size of the IEEE 802.15.4 standard. The TIR region broadcast is transmitted by the fault border nodes and retransmitted by nodes that are a part of the TIR corrective region. This limits the involvement of nodes to those contained in the corrective region.

One concern relating to the approach is the nature of the function to provide adequate balance for irregular. The diagram in Figure 7.19 shows an irregular failure. The nodes coloured in red have failed. The nodes coloured in orange will identity the failure through address space imbalance and construct a TIR of which they will not be a part. They will still be able to save data but will be unable to take part in storage or routing; this may or may not be a problem depending on the immediate and future stated of those devices. If there is constant failure, those nodes would be destroyed. If not, then there would be an imbalance through their isolation.

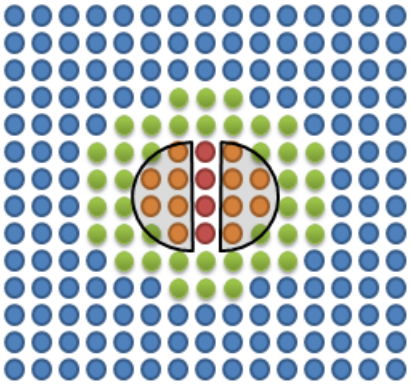


Figure 7.19: Irregular Failure

In Chapter 8, future work will describe the work relating to occlusions to the TIR to address the balancing concerns of irregular failure.

7.5 Conclusion

This chapter details a novel Topological Isomorphic function centred on resolving the load-balancing issues found in GHT-based wireless sensor networks. This chapter details the implementation of a routing protocol that uses Topological Isomorphism to transform a GHT address space to remove the storage and routing load from the border of a fault.

Results provided through simulation have shown that Topological Isomorphic Routing Protocol (TIR) performs well against GPSR, displaying good load-balancing characteristics in both data placement and node route-loading.

Through experimentation, the following conclusions have been reached:

- Topological Isomorphic Functions provide a method to distribute storage and routing load away from failed regions in a Geographical Hash Tables.
- State imbalance when deploying greedy forwarding in wireless sensor networks can be used to identify the border of a fault. Nodes communicating the extent of the fault can seed a distributed function to form a set of nodes to apply topological functions to modify their behaviour when saving and forwarding data.
- The larger the failure, the fewer the number of states mapping to the correcting region around the fault boundary. This could be useful when managing expanding faults.

Future work will explore the border-avoidance strategy to mitigate the impact of an increasing failure region. In cases where an area experiences continued failure, it might be optimal to structure the TIR region in such a way to anticipate the continued failure and accordingly reduce the impact of continually modifying the topology.

Chapter 8

Conclusions and Future Work

This thesis presents novel research relating to the support of communication between Internet of Things (IoT) devices. A number of problems have been identified that will arise from the deployment of the billions of wireless devices that will form the Internet of Things. A number of novel advancements to current research have been provided to support the capability of devices to communicate when connected to the Internet and also to maintain communication when connectivity to the wider Internet is lost. The requirement of device state preservation has been identified, supporting devices to redundantly store their internal state to protect against the possibility that the device is lost due to failure, loss of connectivity or mobility. Architectures have been proposed that detail the way in which future cities might interact with IoT devices that have combined to form smart spaces; these spaces could be within industrial or home residential settings. A primary aim of this research has been to provide the capability for devices to support use cases that require redundant communication to support the application of Internet of Things technology to critical infrastructure, providing better support when these systems are failing.

The information provided by Internet of Thing devices will be invaluable to first responders as they try to mitigate the impact of failure. This is essential to both critical infrastructures and other environments where there is the possibility of loss of life. It continues from the identification of requirements to provide the analysis, design and evaluation of fault-tolerant communication mechanisms to support the requirements of state preservation within failing wireless networks through state rendezvous and Topological Isomorphism. Contributions to the address space balancing of graph layout-based localisation identity schemes have also been provided in this work.

This chapter is organised as follows: Section 8.1 provides a summary of the thesis; Section 8.2 identifies the research contributions made; Section 8.3 describes the future work that will extend the research conducted in this thesis; and Section 8.4 provides concluding remarks.

8.1 Thesis Summary

The Internet of Things is the evolution of wireless sensor network technology from a network of proprietary devices that operate in isolation into a network of devices implementing standards that provide global reachability, enabling smarter digital spaces that have the possibility to be connected together to form future smarter cities. The pervasiveness of the Internet of Things will enable a new level of information-sharing, providing unparalleled insight into the environments surrounding us. Information gathered from the IoT can be provided to higher level systems in an effort to help with automated decision-making, providing new levels of command and control capability as IoT systems will not be limited to sensing and will be capable of modifying the environment. These systems will automate Heating, Healthcare and Security to name but a few supporting concerns. This will provide the capability to optimise the environment resulting in less waste and more personalised services. The pervasiveness of this network will also provide support during times that the environment is experiencing difficulties, such as during a fire, for example. IoT devices can provide additional information to first responders regarding the situation that is currently taking place. During such events, it cannot be taken for granted that devices will be able to maintain their usual connectivity to the Internet; rather, it is expected that these devices should be able to operate in isolation from those high-level systems that usually provide storage and processing capabilities. Instead, devices will need to cooperate to support the distribution and protection of the knowledge contained within the network.

Network knowledge in this context is the environmental knowledge of the Smart Space. For example, during a fire, devices will monitor the environment within the Smart space, persisting their internal state within the system. This can be used during the event to provide additional information to first responders, helping them mitigate the failure, saving more of the physical space from damage, and providing better support to those trapped within. Preserved state within the smart space will be essential following the failure, providing investigators with a detailed account of the event through the observations and internal state of the Internet of Thing devices whose state exists within the remaining system.

This work has focused on the design of protocols to facilitate the communication of device state with the Internet of Things.

- Chapter 1 provided the introduction and initial context for this work and details the way in which sensor networks will be become ubiquitous in the Future Internet, running Internet Protocol stacks and thus enabling greater connectivity and interaction. Moreover, the chapter provides an outline of the requirements that Future Cities and critical infrastructure will place on these devices.

- Chapter 2 provided the reader with a background into the work relating to the deployment of Internet of Things technology and wireless low-power multi-hop environments. This will assist those new to the area of wireless routing in low power

mesh networks to understand the area of work that thesis builds upon. This section also provided the reader with an introduction to the work relating to future cities, identifying the possible impact to the deployment of IoT technology.

- Chapter 3 reviews the work relating to the mobility of devices in respect to the internet and local management of data within low power mesh networks. We found that there was no coherent approach to the management of low power Internet of Things devices and those current approaches would overwhelm the resources of the target devices. The existing schemes look to provide direct communication which would not meet the future city/critical infrastructure requirements of this work.

- Chapter 4 detailed a novel approach to the management of identity and reachability for Internet of Things devices. The approach described uses a combination of overlay and position relative identity to generate a composite key. Correspondents do not use this address to communicate directly; instead, devices in the overlay provide indirection capability with correspondents communicating via the rendezvous point in order to support mobility and to provide protection for the device's state when not in contact with the network. In this work we address the issues relating to the mobility of low power devices that are present in the existing work such as i3 [60]. We also build on the existing work relating to the use of DHT in wireless network, to present a unified architecture. To evaluate this work we use a simulator and real work test bed to identify if the proposed architecture can function within the constraints imposed by low power devices. The results where positive showing the device could work within the constraints imposed by the device and pedestrian mobility patterns. To further improve the reachability of device state within the wireless domain, the chapter identifies the need for an improvement in the distribution of data to prevent loss in the case of network failure.

- Chapter 5 details a novel mechanism to enable distributed rendezvous communication in wireless position relative topologies. The scheme requires no individual device to support its operation. Nodes independently generate key, value tuples to be stored in the DHT space and apply a key rotation mechanism to ensure the separation of data within the network. The approach is evaluated against the random placement that would be observed through the use of overlay or non-geographic position-relative protocols. The approach was evaluated against the traffic placement that would be found when using an overlay DHT, such Virtual ring routing[79]. The results showed that the retention of a devices state was much improved in comparison to the use of overlay schemes.

- Chapter 6. Usually, wireless networks deploying geographic routing utilise devices that are capable of identifying their own coordinates. These devices are commonly equipped with GPS sensors. This approach is difficult to implement as GPS devices typically do not work very well indoors and are expensive. The literature review identifies that the application of graph-drawing algorithms [86] generate good position-relative topologies. However, topologies created with graph-drawing algorithms represent the placement of

devices in a real-world. This does not pose an issue if the devices are evenly distributed inside a square area. In reality, these devices will not be distributed evenly, and the area in which they reside will dictate the shape of the network. To compensate for this, a genetic algorithm with multi variable fitness is applied to the output of the graph-based layout algorithm, to improve the balance of data within the network and improve reachability. We evaluate our approach by testing the proposed protocol against the topology generated by the graph based approach. In an effort to evaluate the approach, a range of topology shapes and sizes are used. In all cases, the reachability of the device state is improved, thus reducing routing overhead when using greedy forwarding alone. This initial topology could become imbalanced following initialisation due to network failure.

- Chapter 7: Having a balanced topology following the network initialisation phase is essential, as is the continued operation of the topology during instances of failure. This chapter explores the use of Topological Isomorphism to adapt Geographic Hash Table address spaces to compensate for localised regular failure. A protocol is defined that can operate in a distributed manor, utilising the address space imbalance to identify those nodes on the perimeter of the fault. Those nodes communicate the extent of the fault and distribute packets to a subset of the network, which will assist in addressing the imbalance and reachability problems of the failing topology. The scheme is evaluated through simulation against GPSR [99], a geographical routing protocol that is used extensively in research.. The approach taken provides a better routing and storage overhead when compared to GPSR. An interesting characteristic is identified where nodes close to the fault in larger failures receive little routing or storage overhead; this could be used to reduce the overhead on nodes that are about to fail, providing them with more time to save their own state within the topology.

8.2 Research Contributions

The purpose of this research was to identify the nature of the interaction between the low-power devices that will constitute the Internet of Things and the Higher Level Future Cities Support Functions, as well as the distributed machine-to-machine interactions that would occur between devices and first responders in the result of system failure. In the process of the design and implementation of the protocols to support this goal, the following research contributions have been made:

1. Proposed a new architecture to support the global machine-to-machine communication between low-power wireless devices. Specifically, the architecture provides for the global reachability of devices, as well as supporting their mobility as they transit between wireless domains. It provides a mechanism to support the state preservation of individual wireless devices whose current wireless domain is disconnected from the Internet and experiencing device failure. This is achieved through a global state rendezvous. This improves on the existing work in this field by reducing the requirement on an active

home node as well as the overhead incurred by schemes such as NEMO [109]. We address the requirements of low power devices that are not supported in other global indirections schemes such as i3 [60] and integrate the capabilities of DHT schemes such as [73] to better account for the local requirement of low power devices.

2. Following the global indirection mechanism to support device mobility, a design for cross-layer routing and storage protocol is provided in an effort to support state preservation within low-power wireless networks. This protocol copes with local failure within the wireless sub-domain and provides supporting mechanisms for independent state discovery by first responders, enabling communication during the failure and state preservation post-failure to support forensic analysis. This improves on the existing work that utilises overlay mechanism such as Virtual ring routing [79] by providing a guaranteed separation in the physical space. This relies on having a GHT topology that is well balanced, the contributions described in 3 and 4 help to provide this requirement.

3. To provide devices within a Geographic Hash Table (GHT) with a suitable identity, the use of evolutionary algorithms is proposed to resolve the Address Space Balancing problem identified in current localisation techniques. The algorithm provides an identity to a device relative to its position within the environment so as to preserve the geo-separation of data. The algorithm minimises the imbalance of the key space allocation provided by graph-based localisation techniques, and also delivers improved reachability characteristics. This scheme improves on the Graph Based [86] techniques that are typically used to provide a likeness of the physical layout of the devices and are not concerned with address balancing.

4. A distributed algorithm to identify and respond to network failures within low-power wireless networks that are utilising Geographical Hash Tables (GHT), as a distributed storage and communication platform. The approach identifies the existence of a failure utilising the storage imbalance that can occur near the border of the failure. It then utilises a Topological Isomorphic function to distribute routing and storage to a defined area surrounding the fault. This scheme improves on [99] through the identification of the fault area and a distribution of the workload to a subset of the remaining nodes. This provides a reduction in the quantity of traffic routing around the failure and the quantity data being saved on perimeter of the failure. We identified that impact of future failure of would also be lessened due to data not being placed on the nodes that border the failure.

8.3 Future Work

The work contained in this thesis will be extended in a number of directions: the first is centred on addressing the issues relating to irregular failure when applying distributed Topological Isomorphic transformation functions; the second challenge is to support the distributed preservation of schema less information.

- Mobility support in GHT for the Internet of Things:

  In Chapter 4, an architecture is introduced to support low-power devices that will make up the majority of devices in the Future Internet of Things. We detailed an approach to support the mobility of devices using the predicted location of the device with reference to the topology. In Chapter 6, we detailed how to construct these topologies in order to provide assurance to devices distributing state with the topology.

  In future work, we would like to explore the mobility of devices in relation to Geographical Hash Tables to support the reachability of devices using global indirect identities. This would require that nodes distributing packets originating from outside the GHT have the capability to identity the geographical position of the device in relation to the topology. It is of no use if the location is the known location of the node; it will need to be the location at which the node will be at the time the packet researches its intended exit point. We envisage that this might involve the modelling of the mobility as this could have an impact on the address space.

- Fault border avoidance using Topological Isomorphic Routing:

  In Chapter 7, we detailed the application of Topological Isomorphism in mind of avoiding the local minimum found when navigating holes found in Geographical Routing Schemes. Through experimentation, we identified that the nature of the topological function resulted in either a reduced load or zero load for those devices on the perimeter of the failure. If a network is experiencing continuous failure, the process of correcting a topology wastes network bandwidth and impacts the level of redundancy provided if finding that, as soon as the corrective action is being initiated, the topology has changed. In future work, we would like to tune the rate of failure to the Topological Isomorphic function to reduce the overhead of establishing Topological Isomorphic region. Taking account of the rate of failure would also provide invaluable to first responders as they could accurately take into account the propagation of failure and accordingly alter their response. The adaptions in the function could also take into account the inefficiencies found when compensating for non-regular failure.

- Reducing Occlusions in Topological Isomorphic Routing:

  The Topological Isomorphic function defined in Chapter 7 creates occlusions when failure is not localised. The nodes that are functioning when we apply the address space transformation become occluded from the topology; they are still able to transmit information to be saved in the topology but will not be able to save information. In order to resolve this, we would like to provide the nodes that have been occluded the capability to alter the shape of the Topological Isomorphic region. It is envisaged that this would take into account the stabilisation of the topology, i.e. no further expiation of the fault. Alternatively, nodes on the border of the fault can generate a function that better describes the fault characteristics or possibly use the combination of the two.

- Schema-less No-SQL data storage in GHT:

  In Chapter 5, we provide a mechanism where devices in wireless GHT networks communicate via a distributed rendezvous function. The nature of the function when applied to balanced GHT topologies provides devices with the assurance that their state will be preserved following a partial failure of the network. The rendezvous function generates multiple identities/locations where the devices state will be placed within the topology. Given that these devices will utilise CoAP to communicate, we seed the generation of identities using the full UDP CoAP [62] service URI of the distributing node. For example, a full service URI for the pressure detected by a device would be coap:// fe80::202:b38e:ac13/pressure. A device that wishes to read the state of another must know this identifier to seed the generation of the key set. If a first responder wanted to identify the states of a number of devices in a building that was on fire, they would need plans annotated with the identities of the device; these could then be used to identify the data distributed by those nodes. This is not ideal, especially when we are dealing with mobile objects, such as gas canisters, humans or even pets. In these situations, responders would need the capability of querying the network for data items that are of concern; this would require that objects have classifications by which they can be grouped. For example, 'in-fire-hazard' could be used for objects that would be dangerous during a fire. The device would distribute information relating to their temperature and relative location. Something attached to a living object could transmit information about wellbeing via a 'wellbeing' key.

  Requiring that a device know the identity of an endpoint in order to communicate with it is not unusual: for example, on the Internet, you either know the IP address of the server with which you wish to communicate or you have the domain name that is translated into the IP address by a look-up service. If you do not know the server identity with which you want to communicate and instead need to find it based on some parameters relating to the content it holds, a discovery service, such as a search engine, is required. Discovery can be provided by an entity of which the identity is known; this does not negate the requirements of knowing an identity of a discovery service. Although in this case it is not the final identity of the

device that holds the information required, rather, it is an intermediary who holds the identities of many devices with a capability to match your requirements.

Alternatively, as found in P2P systems, this look-up can be distributed, which does not require knowledge of a specific identity of a device; rather, the search for information can be propagated using the relationship formed in the overlay space. Applying this technique to the wireless domain would result in flooding queries to all nodes within the system—an undesirable approach that would consume network capacity and resource. No SQL approaches provide a good candidate to store information in a fully distributed way, providing a mechanism to discover information using a schema less approach. For example, a device would push its current state using the rendezvous approach detailed above. The device would additionally store its identity in the overlay using a descriptive key to seed the rendezvous mechanism. The descriptive key could be 'wellbeing'. A first responder could then calculate the key location for 'wellbeing' and retrieve a list of identities that provide information from living things with the area affected. These identities could in turn be queried for their individual sate variables deviceid/heartrate, deviceid/location, etc. The problem arises when multiple living objects key their identities using the 'wellbeingkey', meaning the keys from all nodes would map to the same points. This would cause an imbalance in the databases held on these nodes. In future works, we would like to explore a fully distributed balanced mechanism to the key look-up problem.

8.4 Concluding Remarks

Advances in wireless sensing capability, coupled with the acceptance of technology to monitor our actions, will witness a radical change in the capability of our environment to support those with which it is interacting. The Internet of Things is an enabling technology that will see billions of devices forming individual Smart spaces that will converge to form Future Cities. Individual devices will have global reachability, providing unprecedented levels of information to optimising functions that will help organisations and individuals to meet future challenges. Ensuring the continued operation of these devices in the face of failure will be an essential requirement, especially when we consider the deployment of IoT technology to critical infrastructure. Devices will be expected to operate during adversity through the continued support of machine-to-machine communication when normal communication mechanisms have failed. This continued operation will assist first responders to engage with the IoT smart space in real-time, helping to mitigate the impact of failure. It is also important that IoT-enabled Smart spaces have the capability to support post-failure analysis; this will provide investigators with the opportunity to identify root causes to failure and thus help prevent similar occurrences.

New approaches are required to support the requirements placed on future Internet of Things networks. In this thesis, we identify the challenges that will be placed on the Future Internet of

Things as we look to support critical infrastructure. We present multiple novel techniques that will assist future engineers and researchers in addressing the challenges raised.

The novel solutions detailed in this thesis comprise four interrelated components: (1) A high-level architecture detailing how Future Cities and Internet of Things networks can interoperate to solve cross-cutting concerns; (2) Identification of a mechanism for supporting the global connectivity of IoT devices, compensating for mobility and the sleep state characteristics that will impact on the reachability of low-power devices and thus their capability to interact with Future City functions; (3) The development of a communication mechanism to distribute IoT device state within Geographical Hash Table (GHT)topologies using rendezvous communication; (4) Development of an identity localisation scheme to address the imbalance found in graph-drawing approaches, such as those based on the Fruchterman-Reingold method; and (5) A distributed function to mitigate the impact of failure when deploying rendezvous protocols to Geographical Hash Tables (GHT).

In an attempt to evaluate the techniques suggested, we have utilised simulation to provide us with the scalability required of Future Internet of Things deployments. The evaluation carried out focused on three main characteristics. For the proposed Rendezvous protocol, we were concerned with the ability for devices to distribute its state and for the network to retain the state given a certain percentage of network loss. The results show that the proposed mechanism outperforms the approach of random distribution found in overlay topologies. The Address allocation scheme was measured in terms of its ability to reduce the number of states that were unreachable by nodes in the topology and to reduce the overhead of storage requirement. The approach provides an improvement over the use of Fruchterman-Reingold for all test topology types. For the proposed Topological Isomorphic routing protocol, we address the imbalance that can occur in GHT as a result of greedy forwarding. The approach provides excellent results against the use of GPSR for localised regular failure. Moreover, we have identified that irregular failure will need additional work, as has been outlined in the future work sections as Occlusion for Topological Isomorphic routing.

References

[1] C. E. Perkins and E. M. Belding-Royer, "Ad-hoc On-Demand Distance Vector Routing," in *WMCSA*, 1999, pp. 90–100.

[2] X. Wang and A. O. Lim, "IEEE 802.11s wireless mesh networks: Framework and challenges," *Ad Hoc Networks*, vol. vol, pp. 970–984, 2008.

[3] P. Brenner, "A Technical Tutorial on the IEEE 802.11 Protocol." BreezeCOM Wireless Communications, 1997.

[4] E. M. Royer and C. E. Perkins, "An implementation study of the AODV routing protocol," *2007*.

[5] A. Sgora, D. D. Vergados, and P. Chatzimisios, "IEEE 802.11s wireless mesh networks: Challenges and perspectives," *in MOBILIGHT*, vol. pp, pp. 263–271, 2009.

[6] R. a. Dolin, "Deploying the 'Internet of Things'," *Int. Symp. Appl. Internet*, pp. 216–219, 2006.

[7] A. Cunha, A. Koubaa, R. Severino, and M. Alves, "Open-ZB: an open-source implementation of the IEEE 802.15. 4/ZigBee protocol stack on TinyOS," in *Proc. of the 4th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 07), Pisa, Italy*, 2007.

[8] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," in *HICSS*, 2000.

[9] A. H. Chowdhury, M. Ikram, H.-S. Cha, H. Redwan, S. M. S. Shams, K.-H. Kim, and S.-W. Yoo, "Route-over vs mesh-under routing in 6LoWPAN," in *IWCMC '09: Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing*, 2009, pp. 1208–1212.

[10] G. Mulligan, "The 6LoWPAN architecture," in *Proceedings of the 4th Workshop on Embedded Networked Sensors, EmNets 2007, Cork, Ireland, June 25-26, 2007*, 2007, pp. 78–82.

[11] T. Nicolas, J. Eriksson, and A. Dunkels, "Poster Abstract: Low-Power Wireless IPv6 Routing with ContikiRPL.," in *The 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2010, pp. 12–16.

[12] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, "TinyOS: An operating system for sensor networks," in in *Ambient Intelligence*, W. Weber, J. M. Rabaey, and E. Aarts, Eds. Springer-Verlag, 2005, pp. 115–148.

[13] Z. Shelby and C. Bormann, *6LoWPAN: The Wireless Embedded Internet*. Wiley, 2010.

[14] P. Kinney, “Zigbee technology: Wireless control that simply works,” *Commun. Des. Conf.*, no. October, pp. 1–20, 2003.

[15] D. B. Johnson, D. A. Maltz, and J. Broch, “DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks,” in *In Ad Hoc Networking, edited by Charles E. Perkins, Chapter 5*, 2001, pp. 139–172.

[16] J. Jubin and J. D. Tornow, “The DARPA Packet Radio Network Protocol,” vol. 75, no. 1, pp. 21–32, Jan. 1987.

[17] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, “Optimized Link State Routing Protocol for Ad Hoc Networks.” 2001.

[18] M. Bahr, “Update on the Hybrid Wireless Mesh Protocol of IEEE 802.11s,” in *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on*, 2007, pp. 1–6.

[19] S. Mueller, R. Tsang, and D. Ghosal, “Multipath routing in mobile ad hoc networks: Issues and challenges,” in *In Performance Tools and Applications to Networked Systems, volume 2965 of LNCS*, 2004, pp. 209–234.

[20] M. Grossglauser and D. N. C. Tse, “Mobility increases the capacity of ad hoc wireless networks,” *IEEE\ACM Trans. Netw.*, vol. 10, no. 4, pp. 477–486, Aug. 2002.

[21] T. Winter and P. Thubert, “RPL: IPv6 Routing Protocol for Low power and Lossy Networks.,” *Internet Draft Draft. Work Prog.*, 2010.

[22] D. Berry, A. Usmani, J. L. Torero, A. Tate, S. McLaughlin, S. Potter, A. Trew, R. Baxter, M. Bull, and M. Atkinson, “FireGrid: Integrated emergency response and fire safety engineering for the future built environment.” UK e-Science Programme All Hands Meeting, 20-Sep-2005.

[23] R. A. Naja Iman , Moreau Luc, “Provenance of Decisions in Emergency Response Environments,” *Lect. Notes Comput. Sci.*, vol. 6378, pp. 221–230, 2010.

[24] B. Shafiq, J. Vaidya, V. Atluri, and S. A. Chun, “UICDS compliant resource management system for emergency response,” pp. 23–31, May 2010.

[25] M. Naphade, G. Banavar, C. Harrison, J. Paraszczak, and R. Morris, “Smarter Cities and Their Innovation Challenges,” *Computer (Long. Beach. Calif).*, vol. 44, no. 6, pp. 32–39, Jun. 2011.

[26] S. Talwar, K. Johnsson, N. Himayat, and K. D. Johnson, “M2M: From mobile to embedded internet,” *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 36–43, Apr. 2011.

[27] C. Harrison, B. Eckman, R. Hamilton, P. Hartswick, J. Kalagnanam, J. Paraszczak, and P. Williams, “Foundations for Smarter Cities,” *IBM J. Res. Dev.*, vol. 54, no. 4, pp. 1–16, Jul. 2010.

[28] G. Kortuem, F. Kawsar, D. Fitton, and V. Sundramoorthy, “Smart objects as building blocks for the Internet of things,” *IEEE Internet Comput.*, vol. 14, no. 1, pp. 44–51, Jan. 2010.

[29] J. W. Hui and D. E. Culler, “Extending IP to Low-Power, Wireless Personal Area Networks,” *IEEE Internet Comput.*, vol. 12, no. 4, pp. 37–45, Jul. 2008.

[30] A. P. Castellani, M. Gheda, N. Bui, M. Rossi, and M. Zorzi, “Web Services for the Internet of Things through CoAP and EXI,” in *2011 IEEE International Conference on Communications Workshops (ICC)*, 2011, pp. 1–6.

[31] H. G. and J. P. José M. Hernández-Muñoz, Jesús Bernat Vercher, Luis Muñoz, José A. Galache, Mirko Presser,Luis A, “Smart Cities at the Forefront of the Future Internet,” *Lect. Notes Comput. Sci.*, vol. 6656/2011, pp. 447–462, 2011.

[32] P. Maué and J. Ortmann, “Getting across information communities,” *Earth Sci. Informatics*, vol. 2, no. 4, pp. 217–233, Nov. 2009.

[33] J. A. D. Johnson C. Perkins, “Mobility Support in IPv6,” *RFC 3775*, Jun. 2004.

[34] P. Reinbold and O. Bonaventure, “IP micro-mobility protocols,” *IEEE Commun. Surv. tutorials*, vol. 5, no. 1, pp. 40–56, 2003.

[35] E. Zagari, R. Prado, T. Badan, E. Cardozo, M. Magalhaes, J. Carrilho, A. Berenguel, D. Moraes, T. Dolphine, T. Johnson, and L. Westberg, “Design and Implementation of a Network-Centric Micro-Mobility Architecture,” in *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE*, 2009, pp. 1–6.

[36] R. Koodli, “Fast Handovers for Mobile IPv6,” *RFC 4068*, Jul. 2005.

[37] A. E. Bergh and N. Ventura, “PA-FMIP: a Mobility Prediction Assisted Fast Handover Protocol,” in *Military Communications Conference, 2006. MILCOM 2006. IEEE*, 2006, pp. 1–7.

[38] R. Langar, S. Tohme, and G. Le Grand, “Micro mobile MPLS: a new scheme for micro-mobility management in 3G all-IP networks,” in *Computers and Communications, 2005. ISCC 2005. Proceedings. 10th IEEE Symposium on*, 2005, pp. 301–306.

[39] K. Butler, T. R. Farley, P. McDaniel, and J. Rexford, “A Survey of BGP Security Issues and Solutions,” *Proc. IEEE*, vol. 98, no. 1, pp. 100–122, 2010.

[40] “A Survey on BGP Issues and Solutions,” *Arxiv Prepr. arXiv0907.4815*, 2009.

[41] L. Bao and J. J. Garcia-Luna-Aceves, “Stable energy-aware topology management in ad hoc networks,” *Ad Hoc Networks*, 2009.

[42] R. Ramjee, K. Varadhan, L. Salgarelli, S. R. Thuel, S.-Y. Wang, and T. La Porta, "HAWAII: a domain-based approach for supporting mobility in wide-area wireless networks," *IEEE/ACM Trans. Netw.*, vol. 10, no. 3, pp. 396–410, 2002.

[43] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network Mobility NEMO Basic Support Protocol." Jan-2005.

[44] S. Schmid, L. Eggert, M. Brunner, and J. Quittek, "Towards Autonomous Network Domains," in *INFOCOM*, 2006.

[45] C.-K. Chau, J. Crowcroft, K.-W. Lee, and S. H. Y. Wong, "Inter-domain routing for mobile ad hoc networks," in *MobiArch '08: Proceedings of the 3rd international workshop on Mobility in the evolving internet architecture*, 2008, pp. 61–66.

[46] A. J. Ford, S. H. Y. Wong, and C. K. Chau, "Inter-and Intra-Domain Routing Interactions for MANETs," in *Proceedings of the Second Annual Conference of the International Technology Alliance*, 2008, vol. 1, pp. 1–3.

[47] A. I. T. Rowstron and P. Druschel, "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems," in *Middleware 2001, IFIP/ACM International Conference on Distributed Systems Platforms (3rd Middleware'01)*, 2001, vol. 2218, pp. 329–350.

[48] K. Xie, V. W. S. Wong, and V. C. M. Leung, "Support of Micro-Mobility in MPLS-Based Wireless Access Networks," *IEICE Trans.*, vol. 88-B, no. 7, pp. 2735–2742, 2005.

[49] T. Aoyama, "A new generation network: beyond the internet and NGN," *IEEE Commun. Mag.*, vol. 47, no. 5, pp. 82–87, 2009.

[50] X. Xu and D. Guo, "Hierarchical Routing Architecture (HRA)," *Next Gener. Internet Networks, 2008. NGI 2008*, pp. 92–99, 2008.

[51] O. Hanka, C. Spleiß, G. Kunzmann, and J. Eberspächer, "A Novel DHT-Based Network Architecture for the Next Generation Internet," in *The Eighth International Conference on Networks, ICN 2009, 1-6 March 2009, Gosier, Guadeloupe, France*, 2009, pp. 332–341.

[52] C. J. Martinez, D. K. Pandya, and W. M. Lin, "On designing fast nonuniformly distributed IP address lookup hashing algorithms," *IEEE/ACM Trans. Netw.*, vol. 17, no. 6, pp. 1916–1925, 2009.

[53] H. Pucha, S. M. Das, and Y. C. Hu, "Ekta: An Efficient {DHT} Substrate for Distributed Applications in Mobile Ad Hoc Networks," in *WMCSA*, 2004, pp. 163–173.

[54] C. Cramer and T. Fuhrmann, "ISPRP: a message-efficient protocol for initializing structured P2P networks," in *IPCCC*, 2005, pp. 365–370.

[55] M. Pinheiro, S. Sampaio, F. Vasques, and P. Souto, “A DHT-based approach for path selection and message forwarding in IEEE 802.11 s industrial wireless mesh networks,” in *Proceedings of the 14th IEEE international conference on Emerging technologies & factory automation*, 2009, pp. 1021–1030.

[56] M. Menth, M. Hartmann, P. Tran-Gia, and D. Klein, “Future Internet Routing: Motivation and Design Issues (Routing im Internet der Zukunft: Hintergr{ü}nde und Gestaltungsans{ä}tze),” *it - Inf. Technol.*, vol. 50, no. 6, pp. 358–375, 2008.

[57] Lakshminarayanan, Adkins, Perrig, and Stoica, “Towards a Secure Indirection Infrastructure (short),” in *PODC: 23th ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, 2004.

[58] M. Menth M. Hartmann and D. Klein, “Global Locator, Local Locator, and Identifier Split (GLI-Split),University of Warzburg, Institute of Computer Science, Technical Report No. 470, Apr. 2010.”

[59] F. Al-Shraideh, “Host Identity Protocol,” in *ICN/ICONS/MCL*, 2006, p. 203.

[60] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, “Internet indirection infrastructure,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 4, p. 73, 2002.

[61] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, “Chord: A scalable peer-to-peer lookup service for internet applications,” *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 149–160, 2001.

[62] Z. Shelby, K. Hartke, and C. Bormann, “Constrained Application Protocol (CoAP),” 2013.

[63] J. Laffont, J. Marcus, S. Rey, P. Triole, “Internet Peering,” *JSTOR Am. Econ. Rev. Vol. 91, No. 2 (May, 2001), pp. 287-291*.

[64] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, “A scalable content-addressable network,” in *SIGCOMM ’01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, 2001, pp. 161–172.

[65] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph, “Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing.” 2001.

[66] Maymounkov and Mazieres, “Kademlia: A Peer-to-Peer Information System Based on the XOR Metric,” in *International Workshop on Peer-to-Peer Systems (IPTPS), LNCS*, 2002, vol. 1.

[67] D. Malkhi, M. Naor, and D. Ratajczak, “Viceroy: a scalable and dynamic emulation of the butterfly,” in *PODC*, 2002, pp. 183–192.

[68] D. I. Wolinsky, P. St. Juste, P. O. Boykin, and R. Figueiredo, “Addressing the P2P Bootstrap Problem for Small Overlay Networks,” in *2010 IEEE Tenth International Conference on Peer-to-Peer Computing (P2P)*, 2010, pp. 1–10.

[69] I. Banicescu and S. F. Hummel, “Balancing Processor Loads and Exploiting Data Locality in N-Body Simulations,” *Computer (Long. Beach. Calif).*, 1995.

[70] E. Baccelli, T. Zahn, and J. Schiller, “DHT-OLSR.” HAL - CCSd - CNRS, 2007.

[71] G. Yang, L. Chen, T. Sun, B. Zhou, and M. Gerla, “Ad-hoc Storage Overlay System (ASOS): A Delay-Tolerant Approach in MANETs,” no. 0335302, pp. 1–10.

[72] A. Attwood, M. Merabti, and O. Abuelmaatti, “IoMANETs: Mobility architecture for wireless M2M networks,” in *2011 IEEE GLOBECOM Workshops (GC Wkshps)*, 2011, pp. 399–404.

[73] J. Eriksson, M. Faloutsos, and S. V Krishnamurthy, “DART: dynamic address routing for scalable ad hoc and mesh networks,” *IEEE/ACM Trans. Netw.*, vol. 15, no. 1, pp. 119–132, 2007.

[74] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker, “GHT: a geographic hash table for data-centric storage,” in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications - WSNA ’02*, 2002, p. 78.

[75] B. Zhao, Y. Wen, and H. Zhao, “KDSR: An Efficient DHT-Based Routing Protocol for Mobile Ad Hoc Networks,” in *Hybrid Intelligent Systems, 2009. HIS ’09. Ninth International Conference on*, 2009, vol. 2, pp. 245–249.

[76] Y. C. Hu, S. M. Das, and H. Pucha, “Exploiting the Synergy between Peer-to-Peer and Mobile Ad Hoc Networks,” in *Proceedings of Hot OS’03: 9th Workshop on Hot Topics in Operating Systems, May 18-21, 2003, Lihue (Kauai), Hawaii, USA*, 2003, pp. 37–42.

[77] H. Pucha, S. M. Das, and Y. C. Hu, “Ekta: an efficient DHT substrate for distributed applications in mobile ad hoc networks,” in *Mobile Computing Systems and Applications, 2004. WMCSA 2004. Sixth IEEE Workshop on*, 2004, pp. 163–173.

[78] F. Delmastro, “From Pastry to CrossROAD: CROSS-Layer Ring Overlay for AD Hoc Networks,” *Pervasive Comput. Commun. Work. IEEE Int. Conf.*, vol. 0, pp. 60–64, 2005.

[79] M. Caesar, M. Castro, E. B. Nightingale, G. O’Shea, and A. Rowstron, “Virtual Ring Routing : Network Routing Inspired by DHTs,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, p. 351, 2006.

[80] B. Ahlgren, L. Eggert, A. Feldmann, A. Gurtov, and T. R. Henderson, Eds., “Naming and Addressing for Next-Generation Internetworks, 29.10. - 01.11.2006,” in *Naming and Addressing for Next-Generation Internetworks*, 2007, vol. 06441.

[81] E. Baccelli, T. Zahn, and J. Schiller, “DHT-OLSR.” HAL - CCSd - CNRS, 2007.

[82] M. Castro, P. Druschel, Y. C. Hu, and A. Rowstron, “Exploiting Network Proximity in Distributed Hash Tables.” 2002.

[83] A. Casteigts, A. Nayak, and I. Stojmenovic, “Communication protocols for vehicular ad hoc networks,” *Wirel. Commun. Mob. Comput.*, 2009.

[84] T. Kamada and S. Kawai, “An algorithm for drawing general undirected graphs,” *Inf. Process. Lett.*, vol. 31, no. 1, pp. 7–15, Apr. 1989.

[85] A. Efrat, D. Forrester, A. Iyer, S. G. Kobourov, C. Erten, and O. Kilic, “Force-directed approaches to sensor localization,” *ACM Trans. Sen. Netw.*, vol. 7, no. 3, pp. 27:1–27:25, Oct. 2010.

[86] S. Nawaz and S. Jha, “A Graph Drawing Approach to Sensor Network Localization,” in *2007 IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems*, 2007, pp. 1–12.

[87] A. A.-B. Al-Mamou and H. Labiod, “ScatterPastry: An Overlay Routing Using a DHT over Wireless Sensor Networks,” in *The 2007 International Conference on Intelligent Pervasive Computing (IPC 2007)*, 2007, pp. 274–279.

[88] D. A. Bader, V. Agarwal, and K. Madduri, “On the Design and Analysis of Irregular Algorithms on the Cell Processor : A Case Study of List Ranking ∗ Georgia Institute of Technology,” *Cell*, 2007.

[89] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker, “GHT,” in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications - WSNA ’02*, 2002, p. 78.

[90] A. Awad, C. Sommer, R. German, and F. Dressler, “Virtual Cord Protocol (VCP): A flexible DHT-like routing service for sensor networks,” in *2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, 2008, pp. 133–142.

[91] C. E. Perkins and P. Bhagwat, “Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 24, no. 4, pp. 234–244, Oct. 1994.

[92] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu, “Data-Centric Storage in Sensornets with GHT, a Geographic Hash Table,” *Mob. Networks Appl.*, vol. 8, no. 4, pp. 427–442, Aug. 2003.

[93] D. Niculescu and B. Nath, “Ad hoc positioning system (APS),” in *GLOBECOM’01. IEEE Global Telecommunications Conference (Cat. No.01CH37270)*, 2001, vol. 5, pp. 2926–2931.

[94] S. H. Chagas, J. B. Martins, and L. L. de Oliveira, “Genetic Algorithms and Simulated Annealing optimization methods in wireless sensor networks localization using

artificial neural networks," in *2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2012, pp. 928–931.

[95] T. M. J. Fruchterman and E. M. Reingold, "Graph drawing by force-directed placement," *Softw. Pract. Exp.*, vol. 21, no. 11, pp. 1129–1164, Nov. 1991.

[96] S. Nawaz and S. Jha, "A Graph Drawing Approach to Sensor Network Localization," *IEEE Int. Conf. Mob. Adhoc Sens. Syst. Conf.*, vol. 0, pp. 1–12, 2007.

[97] Q. Zhang, J. Wang, C. Jin, J. Ye, C. Ma, and W. Zhang, "Genetic Algorithm Based Wireless Sensor Network Localization," in *2008 Fourth International Conference on Natural Computation*, 2008, vol. 1, pp. 608–613.

[98] H. S. J. U. Evangelos Kranakis, "Compass Routing on Geometric Networks."

[99] B. Karp and H. T. Kung, "GPSR," in *Proceedings of the 6th annual international conference on Mobile computing and networking - MobiCom '00*, 2000, pp. 243–254.

[100] B. Carbunar, A. Grama, and J. Vitek, "Distributed and dynamic voronoi overlays for coverage detection and distributed hash tables in ad-hoc networks," in *Proceedings. Tenth International Conference on Parallel and Distributed Systems, 2004. ICPADS 2004.*, pp. 549–556.

[101] M. E. Renda, G. Resta, and P. Santi, "Load Balancing Hashing in Geographic Hash Tables," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 8, pp. 1508–1519, Aug. 2012.

[102] R. Sarkar, W. Zeng, J. Gao, and X. D. Gu, "Covering space for in-network sensor data storage," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks - IPSN '10*, 2010, p. 232.

[103] J. Gao, F. Li, and Y. Wang, "Distributed Load Balancing Mechanism for Detouring Routing Holes in Sensor Networks," in *2012 IEEE Vehicular Technology Conference (VTC Fall)*, 2012, pp. 1–5.

[104] R. Guleria and A. Kumar Jain, "Geographic Load Balanced Routing in Wireless Sensor Networks," *I. J. Comput. Netw. Inf. Secur.*, no. 8, pp. 62–70, 2013.

[105] L. Popa, A. Rostamizadeh, R. Karp, C. Papadimitriou, and I. Stoica, "Balancing traffic load in wireless networks with curveball routing," in *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing - MobiHoc '07*, 2007, p. 170.

[106] A. Awad, C. Sommer, R. German, and F. Dressler, "Virtual Cord Protocol (VCP): A flexible DHT-like routing service for sensor networks," in *2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, 2008, pp. 133–142.

[107] G. Csardi and T. Nepusz, "The igraph Software Package for Complex Network Research," *InterJournal*, vol. Complex Sy, 2006.

[108] Q. Fang, J. Gao, and L. J. Guibas, “Locating and Bypassing Holes in Sensor Networks,” *Mob. Networks Appl.*, vol. 11, no. 2, pp. 187–200, Mar. 2006.

[109] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, “Network Mobility NEMO Basic Support Protocol.” Jan-2005.

Appendix A:Pseudo code for EB-DHT

```
IF NodeInLens THEN
        IF DataNotInLens THEN
                IF DataDestination - FaultCentre<NodePosition - FaultCentre THEN
                        Left = call RotateLeft(DataDestination)
                        Right = call RotateRight(DataDestination)
                        IF RotateRight<RotateLeft THEN
                                DataDestination = Right
                        ELSE
                                DataDestination = Left
                        ENDIF
                ENDIF
        ENDIF
        FOR Neighbour in Neighbours
                IF Neighbour = SendingId AND NeighbourNotInLens THEN
                        continue
                ENDIF
                IF NodeInLens and DataDetinationInLens THEN
                        IF AngleBetweenDataAndNeighbour> 30 THEN
                                        NeighBourDistance = AngleBetweenDataAndNeighbour
                        ELSE
                                        NeighbourDistance = DataDestination - Neighbour.Address
                        ENDIF
                ELSE
                        NeighbourDistance = DataDestination - Neighbour.Address
                ENDIF

                IF NeighbourDistance<BestTotal
                        BestTotal = NeighbourDistance
                        BestNeighbourId = Neighbour.Id
                ENDIF

ELSE
        FOR Neighbour in Neighbours
                If Neighbour.NotAlive
                        continue
                ENDIF
                        NeighbourDistance = Neighbour.position - DetaDestination
                IF NeighbourDistance<BestTotal
                        BestTotal = NeighbourDistance
                        BestNeighbourId = Neighbour.Id
                ENDIF
        ENDFOR
ENDIF

IF NodeNotInLens THEN
        NodeTotal = NodePosititon - DataDestination
ELSE
        IF DataInLens THEN
                Angle = call CalcAngle(NodePosition,DataDestination)
                IF Angle > 30 THEN
                        NodeTotal = angle
                ELSE
                        NodeTotal = NodePosition - DataDestination
```

```
                ENDIF
        ELSE
                NodeTotal = NodePosition - DataDestination
        ENDIF
ENDIF

IF NodeTotal<= BestTotal THEN
        return ThisNodeId
ELSE
        return BestNeighbourId
ENDIF
```

Appendix B: Genetic Algorithms Graphs

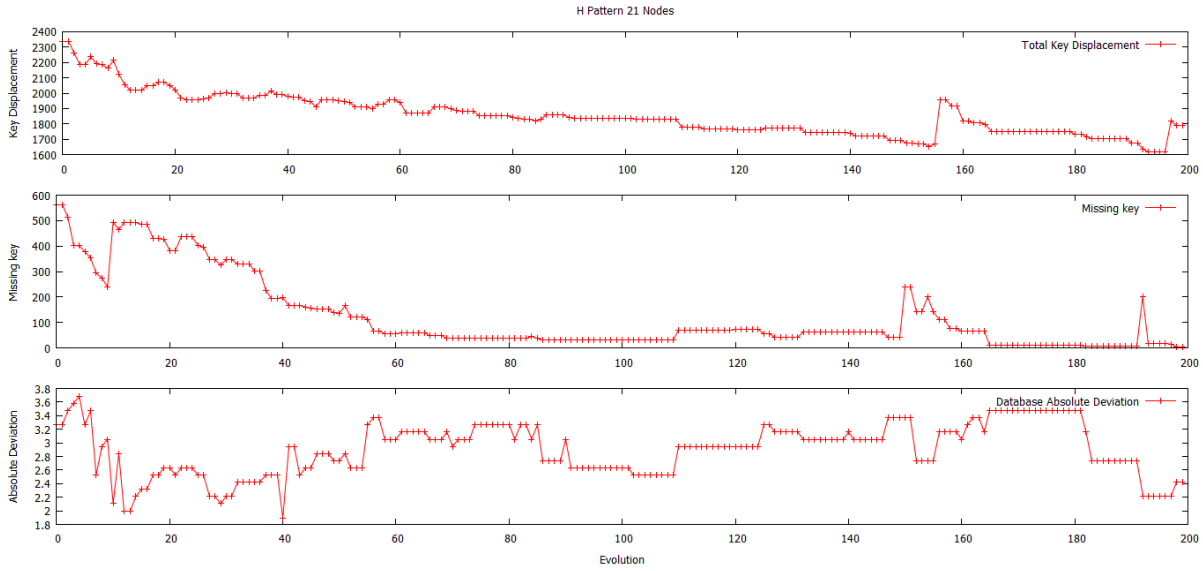


Figure B.1: H Pattern 21 Nodes

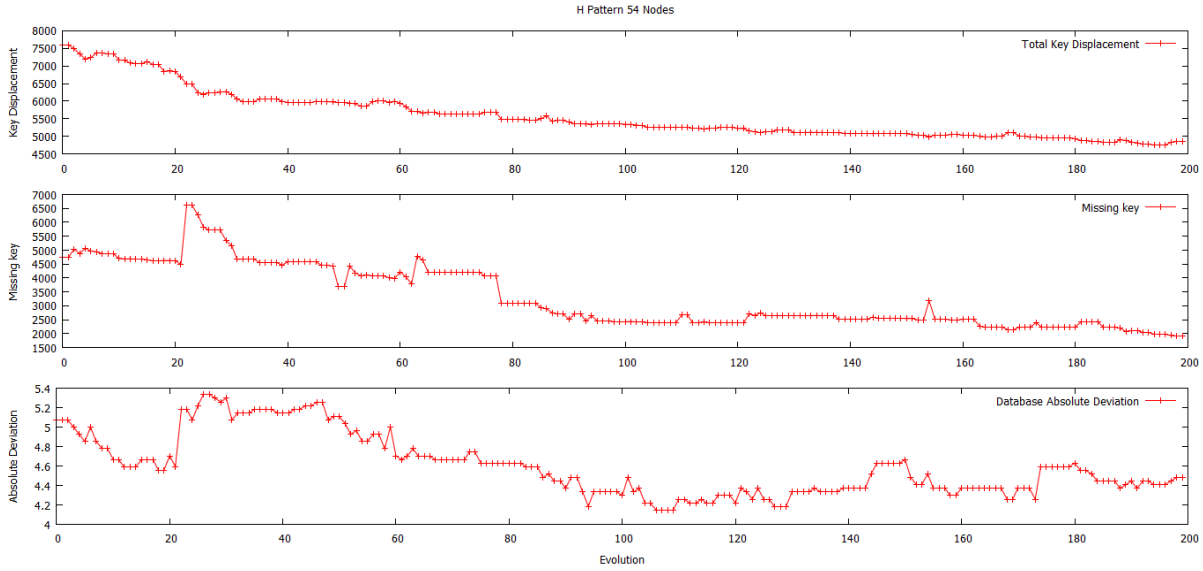


Figure B.2: H Pattern 54 Nodes

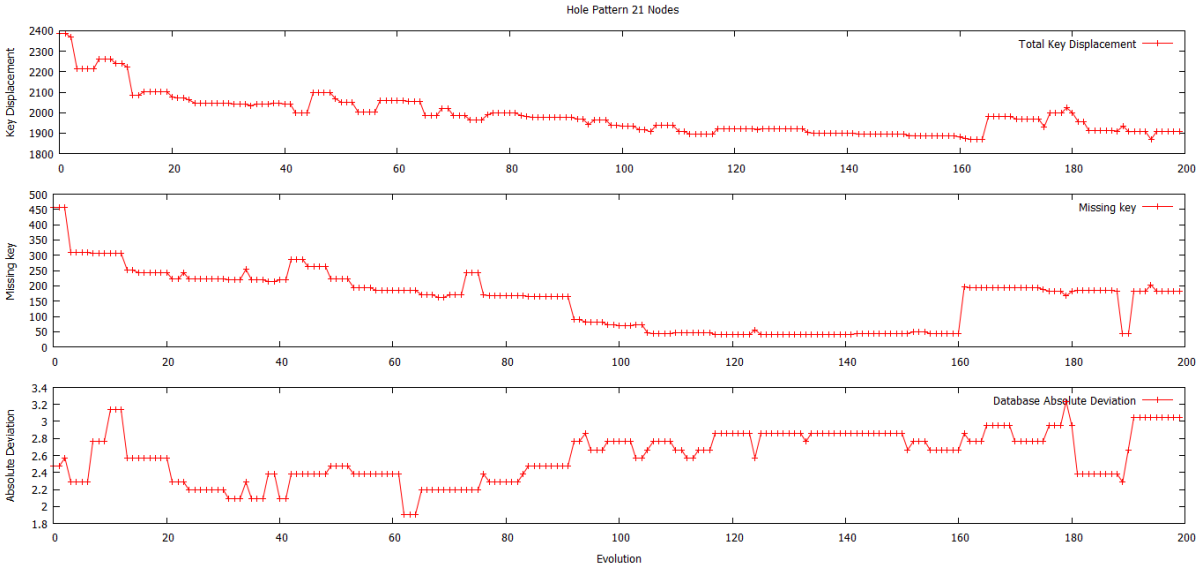


Figure B.3: Hole Pattern 21 Nodes

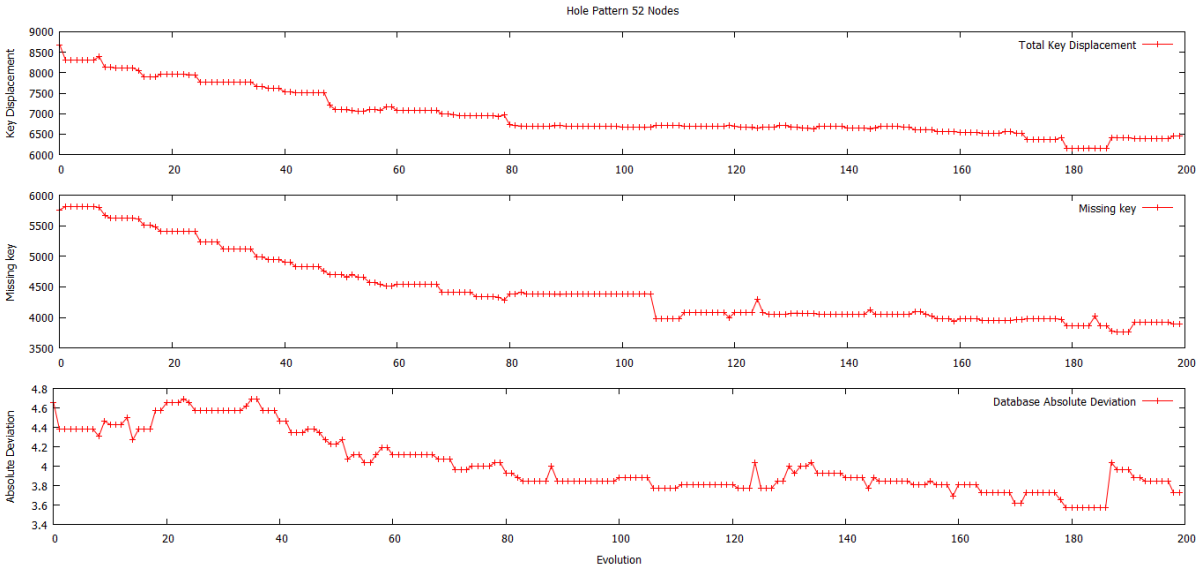


Figure B.4: Hole Pattern 52 Nodes

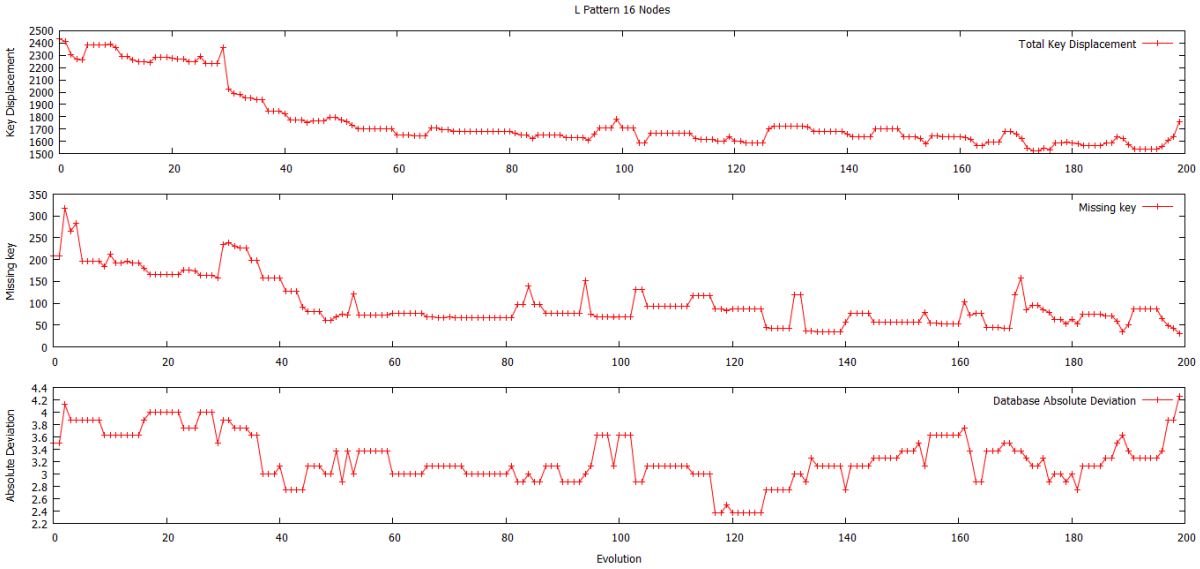


Figure B.5: L Pattern 16 Nodes

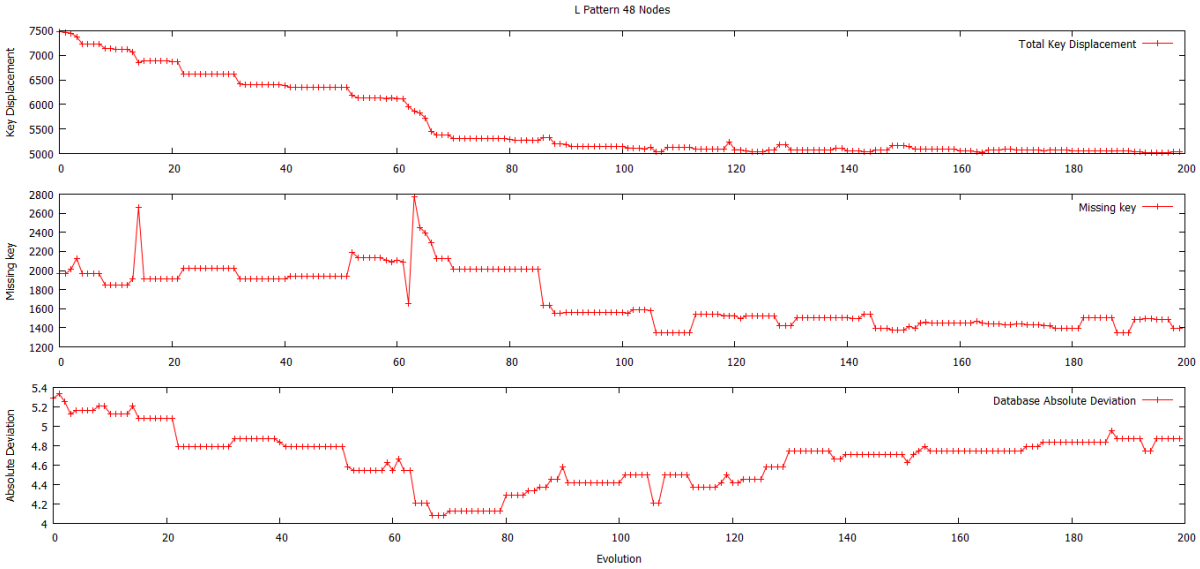


Figure B.6: L Pattern 48 Nodes

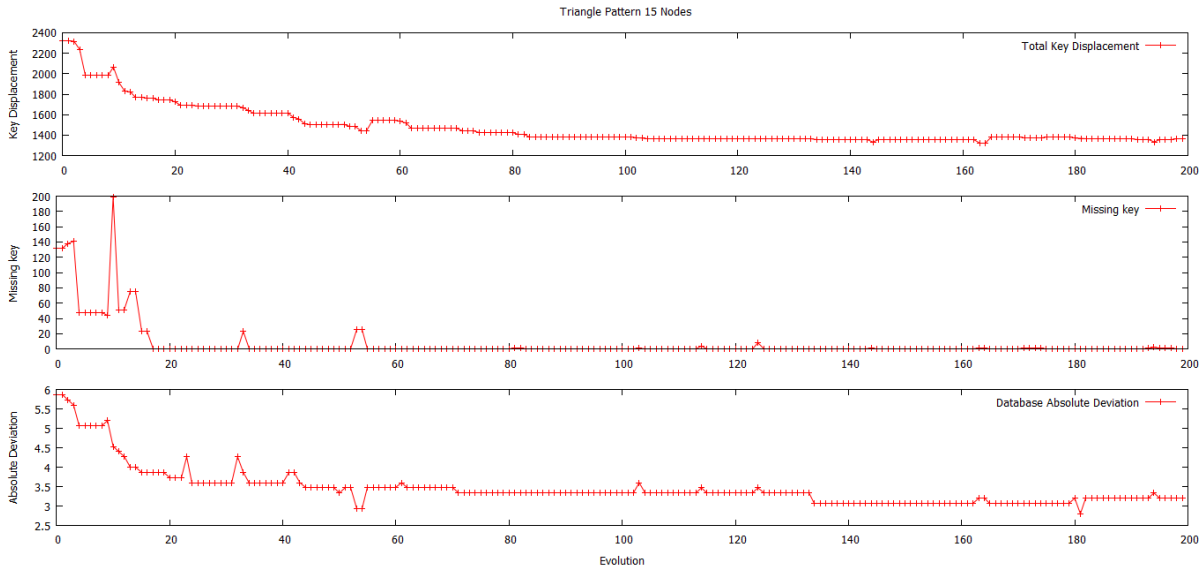


Figure B.7: Triangle Pattern 15 Nodes

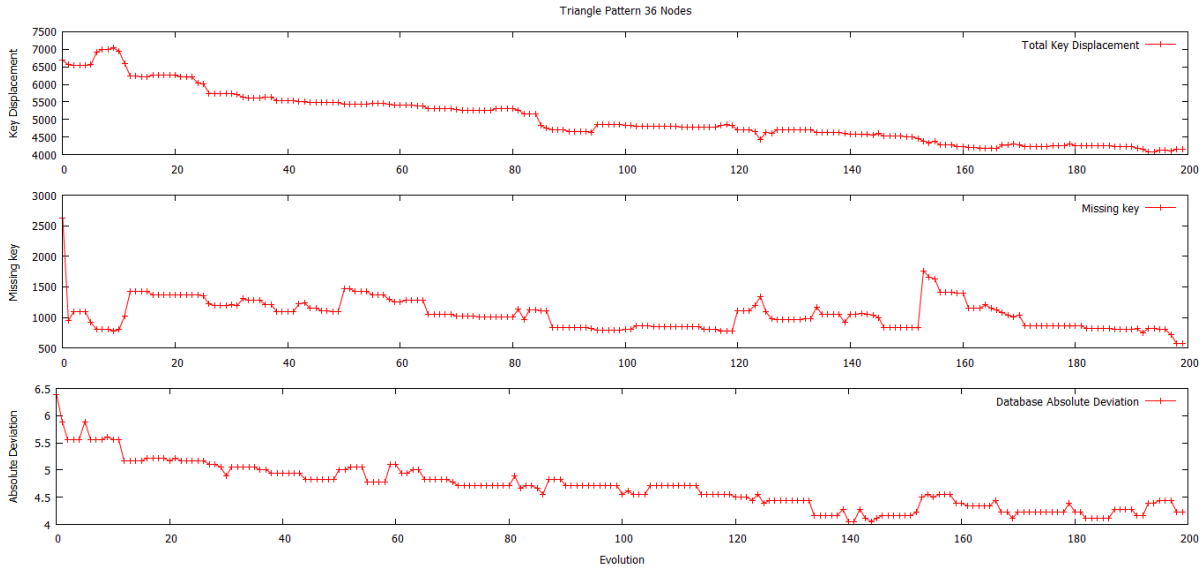


Figure B.8: Triangle Pattern 36 Nodes

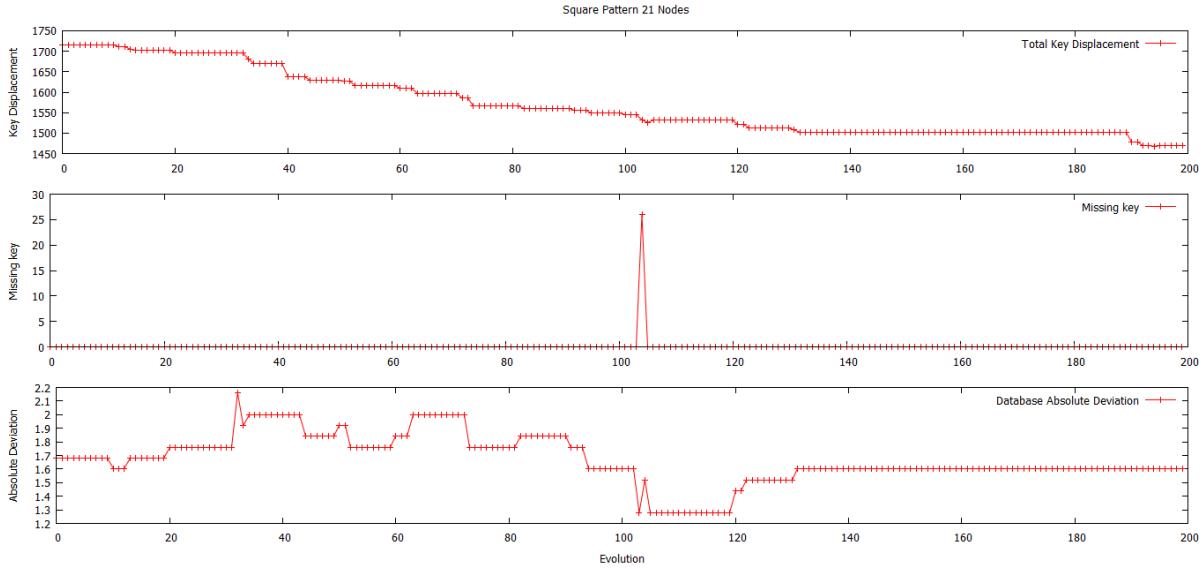


Figure B.9: Square Pattern 21 Nodes

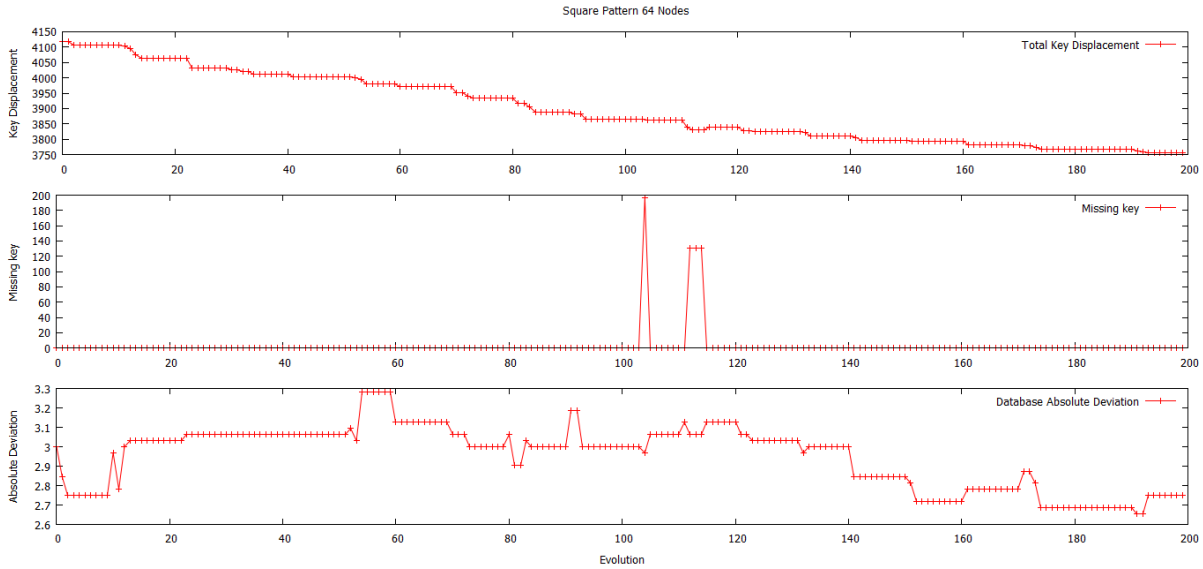
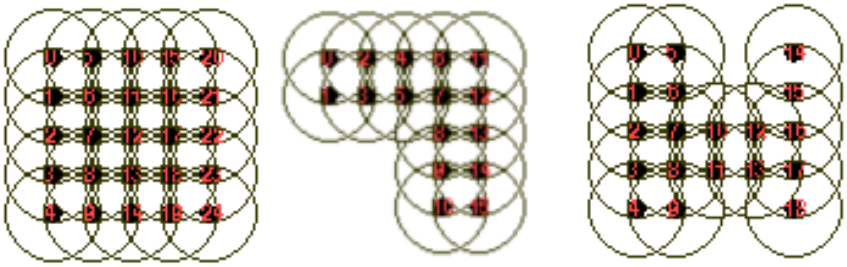
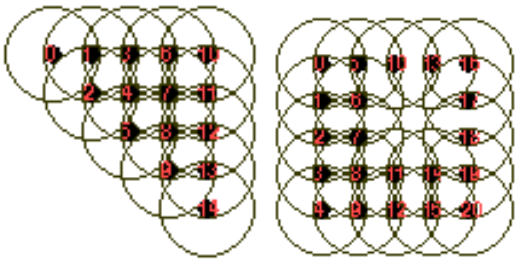


Figure B.10: Square Pattern 64 Nodes

Appendix C: GA Network Test Patterns



Square 25L 16H19



Triangle 15Hole 21

Figure C.1: Small Topology Test Patterns

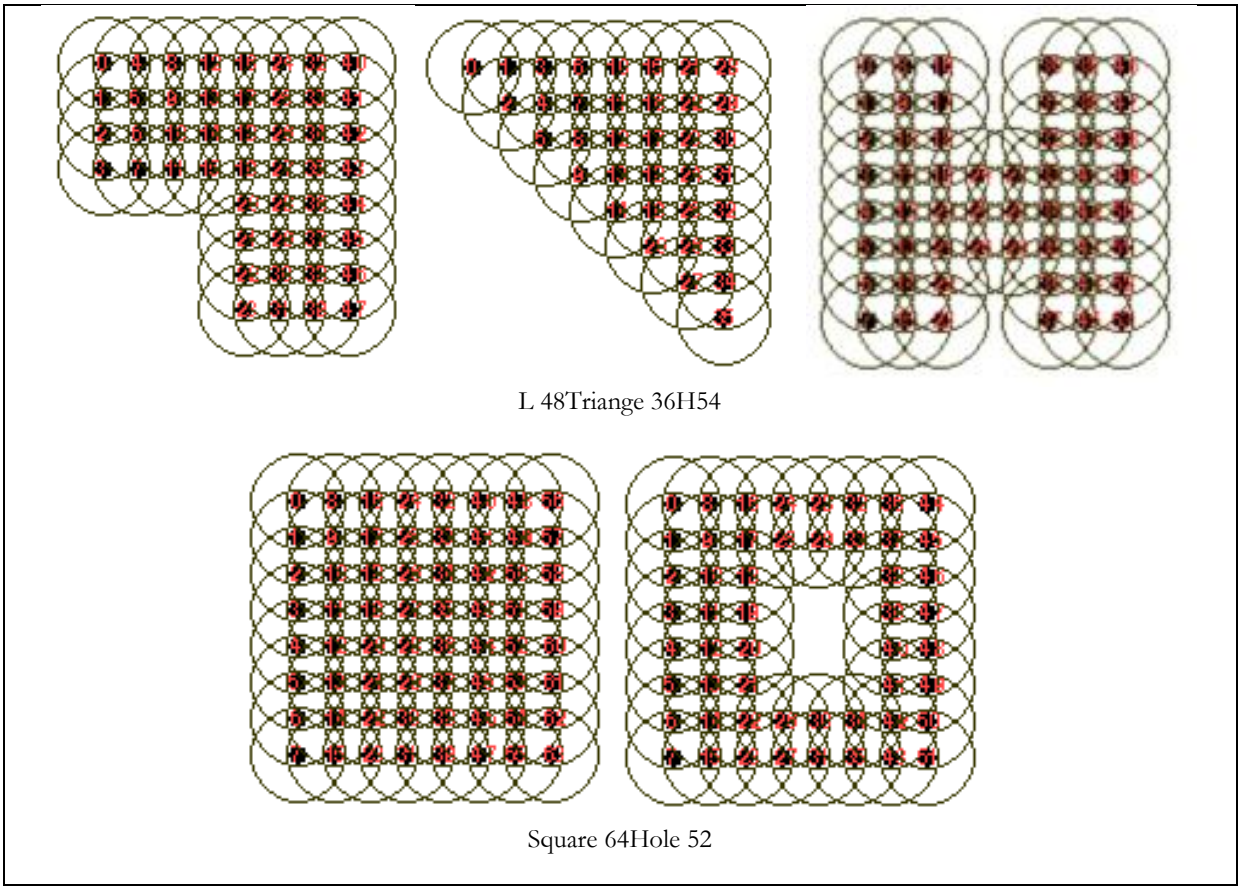
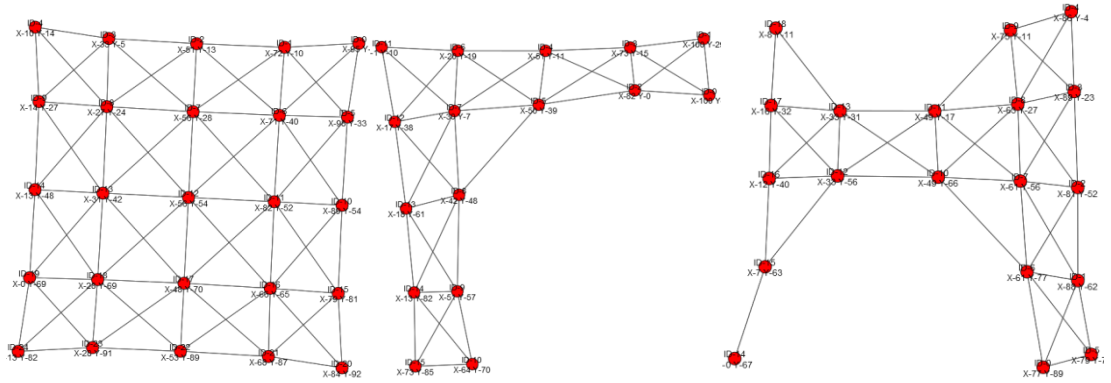
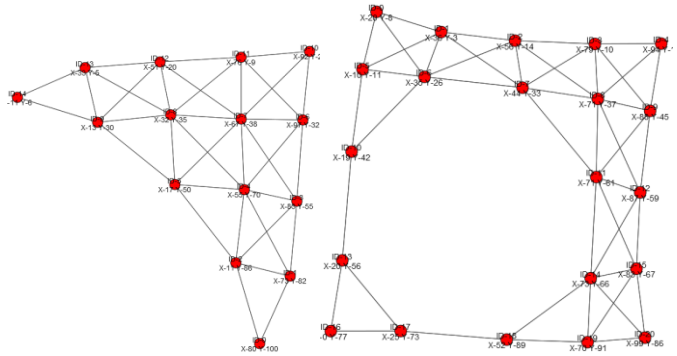


Figure C.2: Large Topology Test Patterns

Appendix D: Fruchterman-Reingold Graphs



Square 25 nodesL 16 nodesH 19 nodes



Triangle 15 nodesHole 21 nodes

Figure D.1: Fruchterman-Reingold Estimation for Small Topologies

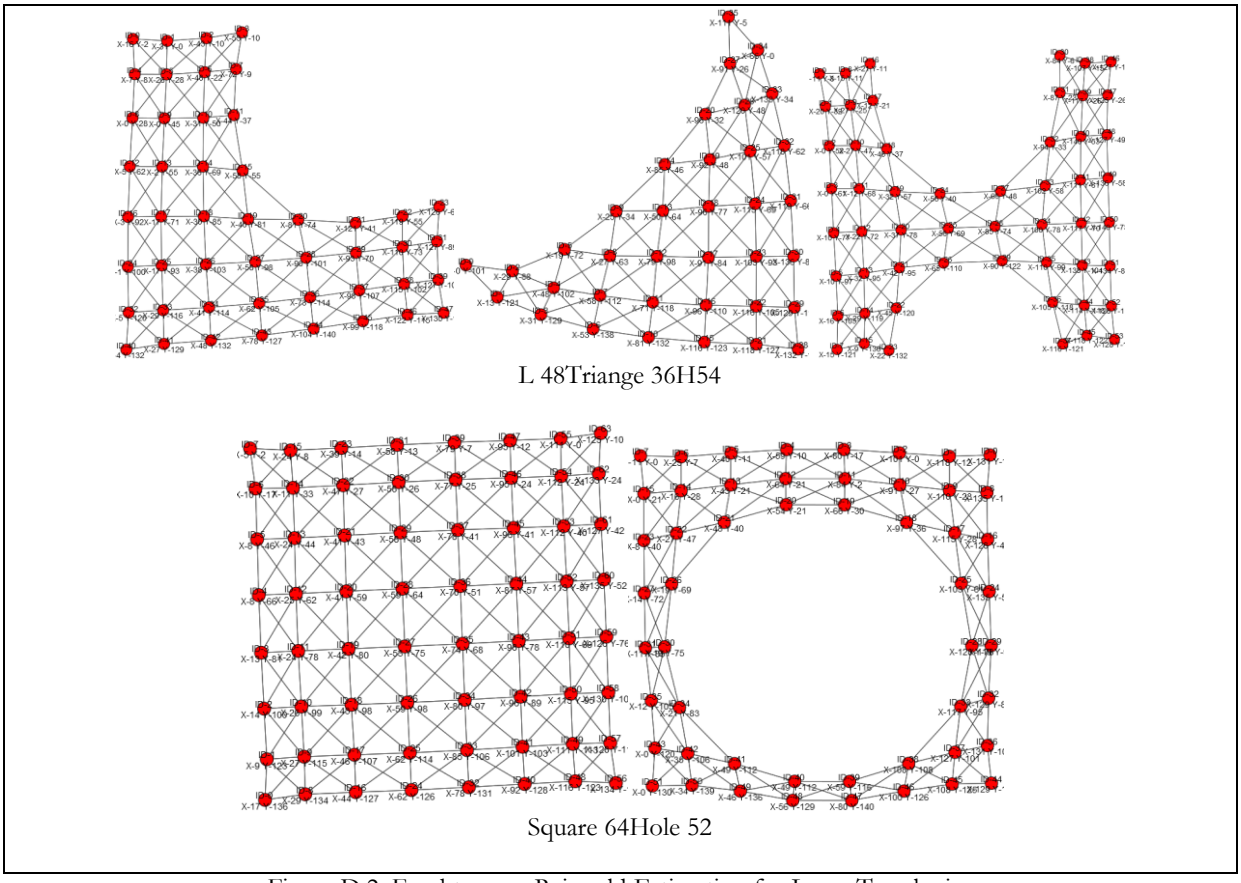
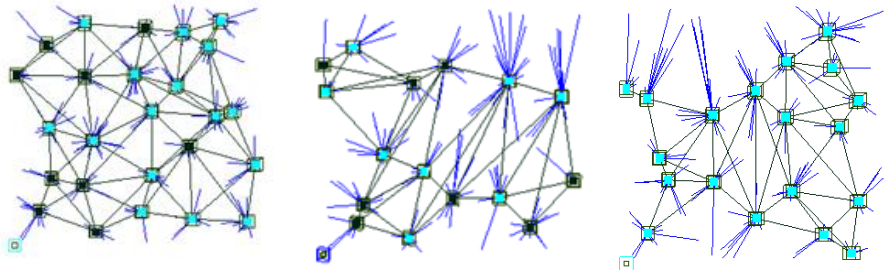
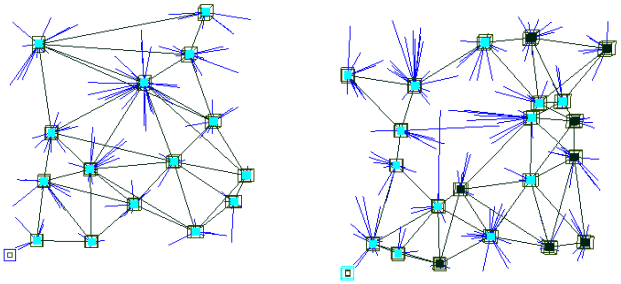


Figure D.2: Fruchterman-Reingold Estimation for Large Topologies

Appendix E: Key Displacement



Square 25L 16H19



Triangle 15Hole 21

Figure E.1: Small Topology Key Displacement Post-GA

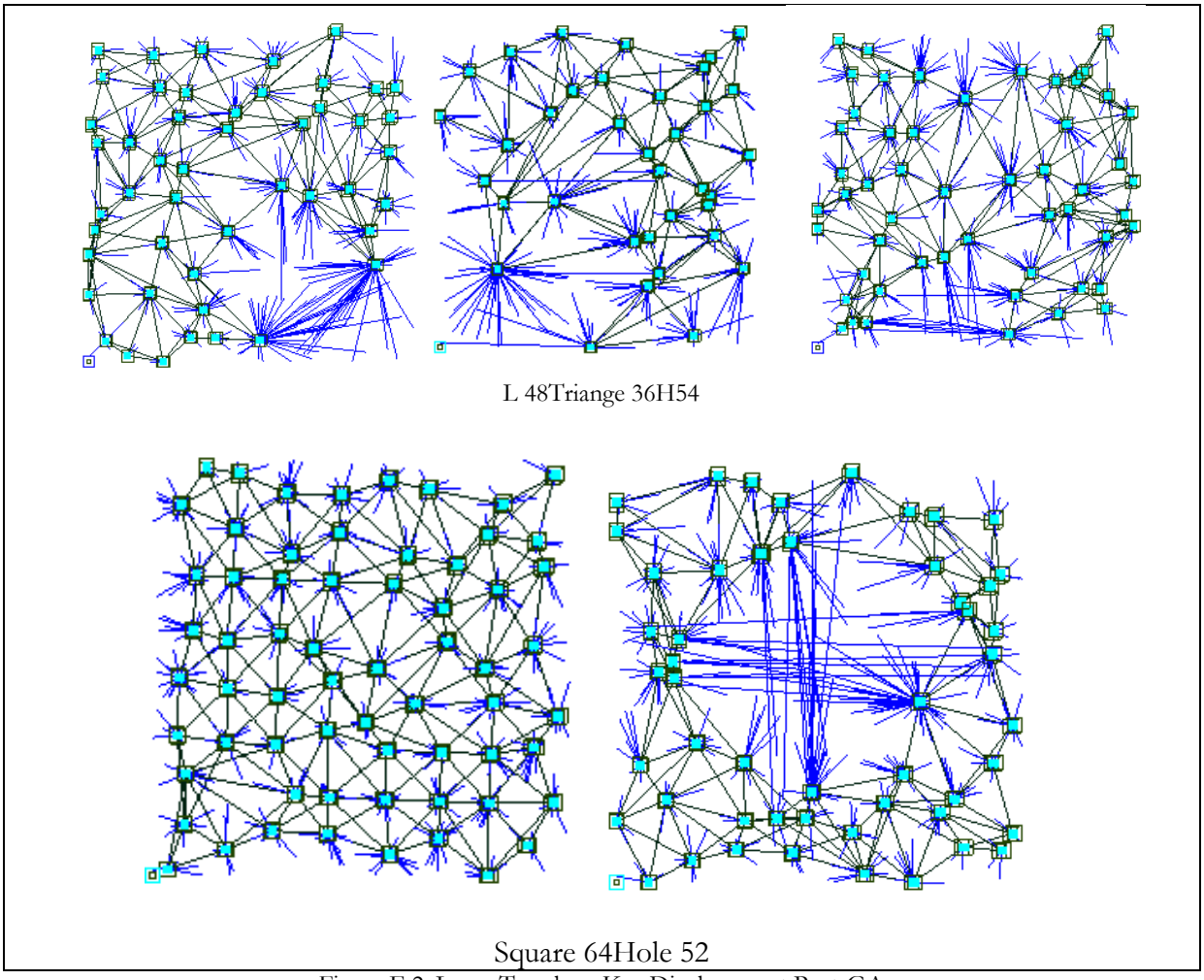
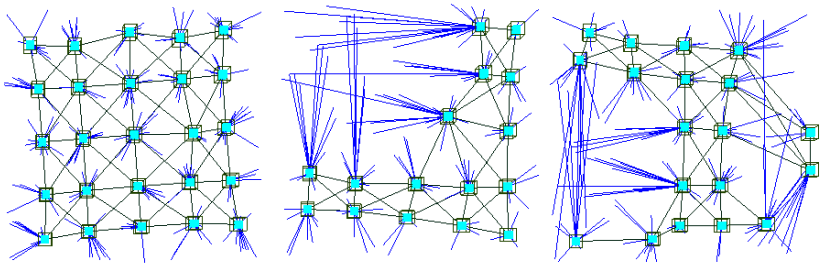
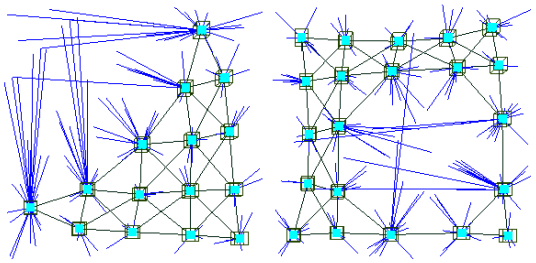


Figure E.2: Large Topology Key Displacement Post-GA

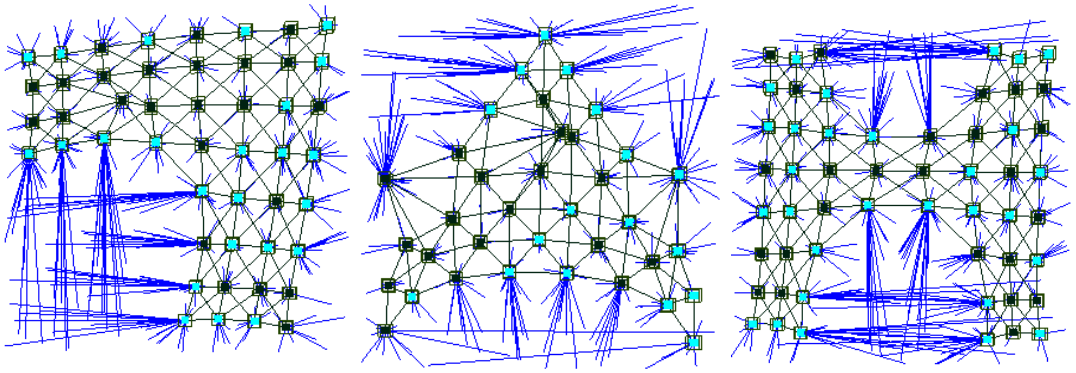


Square 25L 16H19

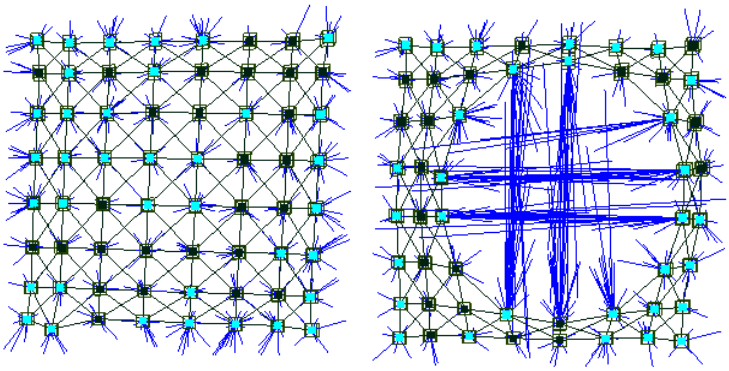


Triangle 15Hole 21

Figure E.3: Small Topology Key Displacement before GA



L 48Triange 36H54



Square 64Hole 52

Figure E.4: Large Topology Key Displacement before GA