VIRTUAL CLOSED NETWORKS: OPTIMISED SECURITY FOR AUTONOMOUS MANETS

DARREN HURLEY-SMITH

A thesis submitted in partial fulfilment of the requirements of the

University of Greenwich for the Degree of Doctor of Philosophy

July 2015

Declaration

I certify that this work has not been accepted in substance for any degree, and is not currently being submitted for any degree other than that of Doctor of Philosophy being studied at the University of Greenwich. I also declare that this work is the result of my own investigations, except where otherwise identified by references, and that I have not plagiarised the work of others.

Student: Darren Hurley-Smith

Signature:

Date:

Supervisor: Dr Jodie Wetherall

Signature:

Date:

Acknowledgements

This work would not have been possible without the support and guidance of key individuals throughout my studies.

First and foremost among these is Dr Jodie Wetherall, who has provided support, understanding and guidance in abundance. He has filled the role of mentor admirably, with great patience and enthusiasm for the research I have undertaken. His guidance and constructive criticism has been invaluable in developing my professional and personal skills throughout the research.

Similarly, Dr Adekunle has provided support and has been an invaluable addition to my supervisory team. He has provided much of his time to debate and discussion on critical points of my work, aiding in my professional development by providing consistent, frequent constructive criticism. His passion for research and unique perspective have been invaluable in galvanising my own enthusiasm for academia.

Dr Woodhead has provided a wealth of experience in many aspects of my journey through the PhD process, providing guidance and encouragement in my research. I'd like to thank all of these individuals for their time as supervisors and the generosity and patience they have shown me with their time and involvement with my work for the past three years.

I would like to thank the University of Greenwich for giving me the opportunity to study as a part of their institution. I would also like to thank my colleagues; Luc Tidy, Khurram Shazad and Muhammad Aminu, for their support and friendship. They have been exemplary colleagues and friends throughout my time at the University of Greenwich.

Finally, I would like to extend my heartfelt thanks to my friends and family for their unconditional love, patience and motivational support, especially my wife Georgina and parents Andrew and Tina Smith.

Abstract

Autonomous mobile platforms (such as Unmanned Aerial Vehicles, also known as UAVs) have become a popular tool in exploration, disaster management, civil-engineering, agricultural and military scenarios. Their endurance, low-cost, high mobility and ability to reduce human involvement in prolonged or hazardous activities have proven attractive to both commercial and military sectors. In such domains, security is required to protect the data, functionality and performance of the network, making it a vital consideration when developing such systems.

Systems capable of independent action, following a human-defined mission without scripting or other forms of direction in the field, are adaptable and effective as a means of achieving individually simple tasks that due to their number and distribution represent complex objectives as a collective. However, such systems must communicate to achieve autonomous function. Efficient distribution of tasks requires significant communication between all members of the network to determine the nodes most fit to undertake a given task. Mobile ad hoc networks (MANETs) provide the foundation for such communication, providing a means by which nodes may communicate with other members of the network, even if they are not in range. Issues arise when considering the security of MANET communication, due to the ease observation, interception and manipulation of data broadcast over such networks. It is trivial for attackers to perform such actions, due to the open nature of the communication medium.

This dissertation presents a novel security framework, which specifically targets autonomous MANET communication. Addressing the open-medium problem by providing a Virtual Closed Network (VCN) environment, Security Using Pre-Existing Routing for MANETs (SUPERMAN) also secures routing and control data, providing confidentiality, integrity and authentication services as a complete solution the network layer and above.

Improvements to the efficiency of communication required by distributed task allocation are proposed (Cluster Form CBBA and Broadcast Enabled Cluster Form CBBA) based on this work, in the interests of optimising the use of network resources to facilitate the addition of robust security measures suitable for resource constrained MANETs.

Index of Acronyms

ACBBA	Asynchronous Consensus Based Bundle Algorithm
AEAD	Authenticated Encryption with Associated Data
AES	Advanced Encryption Standard (Rijndael)
AODV	Ad hoc On-demand Distance Vector
BECF-CBBA	Broadcast Enabled Cluster Form CBBA
CAA	Civilian Aviation Authority (British)
CAP-722	Civilian Aviation Policy Document 722 (living document)
CBBA	Consensus Based Bundle Algorithm
CF-CBBA	Cluster Form Consensus Based Bundle Algorithm
CRT-VSS	Chinese Remainder Theorem Verifiable Secret Sharing
CSMA	Carrier Sense Multiple Access
DDoS	Distributed Denial of Service
DES	Data Encryption Standard
DMG	Diminishing Marginal Gain
DoS	Denial of Service
DTA	Distributed Task Allocation
FAA	Federal Aviation Administration (USA)
GPS	Global Positioning System
HMAC	Hash Message Authentication Code
IKE	Internet Key Exchange
IoT	Internet of Things
IP	Internet protocol
IPsec	Internet Protocol Security
ITU	International Telecommunication Union
LAN	Local Area Network
MANET	Mobile Ad hoc Network
MD5	Message-Digest 5 (hashing algorithm)
MPR	Multi-Point Relay

MTU	Maximum Transmission Unit
NDP	Neighbour Discovery Protocol (IPv6)
OLSR	Optimised Link-State Routing
OSPF-MDR	Open Shortest Path First - MANET Designated Router
PAN	Personal Area Network
RIPSEC	Reputation-Based Internet Protocol Security
SA	Security Association
SSL	Secure Sockets Layer
SUPERMAN	Security Using Pre-Existing Routing for Mobile Ad hoc Networks
SUPERAODV	SUPERMAN Using AODV
SUPEROLSR	SUPERMAN using OLSR
TA	Trusted Authority
T-CBBA	Team Consensus Based Bundle Algorithm
TCP	Traffic Control Protocol
TinyAEAD	A light-weight AEAD implementation
TRUNCMAN	Trust-based Routing Using Non-cooperative Movement in MANETs
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
UGV	Unmanned ground vehicle
USV	Unmanned Submersible vehicle
VCN	Virtual Closed Network
WAN	Wireless Local Area Network
WiFi	Wireless Networking standard 802.11
ZRP	Zone Routing Protocol

Index of Terms

Active Attack	Actions taken by an attacker to damage, destroy or disrupt data or nodes in the network
Authentication	The process by which a node is proven to be a legitimate member of the network
Byte Cost	The number of bytes required to communicate or achieve a solution
CBBA Bundle	A collection of tasks, may be portrayed programmatically as a vector or array
CBBA Optimality	The sum of scores on all CBBA nodes, compared against total achievable, provides a measurement of optimality
CBBA Round	An iteration of CBBA, in which all nodes perform CBBA, but may not arrive at a solution
CBBA Score	The points value given as an indication of task importance, a 'reward' for complation
CBBA Solution	The result of 2 or more rounds, where a validation round finds that no changes need to be made
CBBA Task	A single action, including position, type and associated costs and rewards
Closed Network	A network that does not allow the use of, or interaction with, any untrusted nodes, for any reason. Usually enforced using hardware
Cluster	A collection of nodes, part of a network, but not itself a network. May be subnetted
Communication Complexity	A measure of transmissions required to communicate to a solution, and the costs associated with generating topology to facilitate that
Communication Cost	A general term, used to describe the amount of bytes and individual transmisions required to communicate
Communication Link	An abstract term used to define where a message begins and ends, regardless of other factors
Delegated Authentication	A SUPERMAN process, by which a node may answer on behalf of a node it knows to be authentic, to reduce communication costs
Destination Node	The node intended to receive end-to-end communication

Endpoint	The nodes at the beginning and end of an end-to-end communication link
End-to-end	Communication between two endpoints, regardless of their proximity
Global State	The state of the whole network, often an aggregate of all local states, but may be defined in more abstract terms
Hop	A point-to-point communication, specifically used to refer to movement of data over a route of many nodes
Intermediary	A routing node, on the route between source and destination
Key Derivation	The discovery of elements of a key, by observing multiple instances of its use (usually compared against known, repeated plain text data)
Key Generation	Any process that takes values from two or more nodes to generate a cryptographic key
Key Management	Processes by which keys may be stored, updated, or removed from the network as needed
Key Reuse	Every time a key is used for a unique instance of communication, it is 'reused'
Key-share	A variable (usually a prime number) associated with and communicated by a node during Diffie-Hellman key generation
Link-key	A cryptographic key associated with the relationship between two nodes
Link-wise	Any action that draws on the relationship between two nodes, regardless of their proximity
Local State	A node's current state, regardless of the rest of the network
Mission Area	A designated geographic (or simulated) location in which nodes operate
Mutual Authentication	A process that involves nodes authenticating each other, regardless of who initiated the request for security credentials
Network	A communicating group of nodes
Network Formation	The process (variable) by which a network topology is generated, and nodes linked with each other for the purpose of communication
Network Resources	Bandwidth and queuing buffers, a measure of how much data a link or network can sustain over a given time
Node	A communicating element of a network

Node Identity	A collection of variables that give a node a unique property. Can include address, unique values, capabilities and other identifying features
Node-specific	Information pertaining to an individual, unique node
Offline Attack	A malicious action performed on data derived from interaction with a network, but away from that network
Online Attack	A malicious action performed against data active in the network
Open Network	A network that may employ security countermeasures, but which also allows the integration of untrusted nodes in some capacity
Passive Attack	Observation with malicious intent, where an attacker doesn't act against the network, but may steal data
Point-to-point	Communication between two neighbouring nodes, often when routing between two endpoints
Private Key	A locally stored, non-communicated variable used for asymmetric key generation
Protocol	A series of rules defining the steps taken to perform a process
Public Key	A locally stored, communicated variable used for asymmetric or symmetric key generation
Route	The path taken by a packet, between source and destination
Router	A device that forwards packets in a network or between networks
Security Association	A state where two nodes share keys that allow unique end-to-end encryption
Source Node	The node initiating communication between tow endpoints
Subnet	A collection of addresses within a network, which define a visible logical grouping of nodes by address alone
Task-list	A bundle, usually used to refer to the unmodified, initial bundle allocated to all nodes when preparing to perform CBBA
Trusted Authority	A node (usually not present in the mission area) that is responsible for giving nodes identifying data and security credentials
Virtual Closed Network	A closed network approach that uses homogenous, strict access control policies to prevent involvement of untrusted nodes in MANET communication
Wireless Communication	Communication performed over a medium that does not require physical linking of nodes, such as radio

Wireline Communication	Communication performed between nodes linked by a physical medium, such as copper or optical wire

Contents

Figures

Tables

1 INTRODUCTION

1.1 Chapter Introduction

This thesis has been compiled to summarise the results of research into security solutions suitable for autonomous Mobile Ad hoc Networks (MANETs). The document will begin by identifying and discussing existing literature and the current 'state of the art'. This will allow the identification of original contributions for the thesis.

1.1.1 Chapter Layout

- Section 1.2 provides background information on the central topics of the thesis.
- Section 1.3 outlines research questions that guide the investigation documented in this thesis.
- Section 1.4 outlines broad research objectives based on the research questions, which represent strategic elements of the research.
- Section 1.5 defines the boundary of the research, outlining the point at which the research is considered to be complete and defining out of scope work.
- Section 1.6 states the original contributions encapsulated in the research.
- Section 1.7 provides an overview of the thesis structure, defining each chapter and elaborating briefly on its purpose.
- Section 1.8 summarises this chapter.

1.2 Background

Mobile Ad hoc Networks (MANETs) have become increasingly popular as a means of allowing mobile nodes to communicate and cooperate without the need for traditional network infrastructure, instead treating all nodes as routers and end-points. As a result, MANET nodes are usually required to not only communicate on their own behalf, but

relay messages from other nodes that may be out of range of the target of their transmissions. The means by which this is achieved varies, with many routing protocols proposed and several standardised to date (Lee et al. 1999).

A key consideration with MANETs is that they are mobile and possibly long range in nature, being located far from infrastructure and possibly beyond direct human control. Many unmanned aerial vehicles (UAVs) are remote piloted, either completely or as an assistance measure, but an increasing number of UAV platforms are fully autonomous (McCune & Madey 2013). Swarms of UAVs may exceed the multitasking capabilities of human operators, requiring the development of fully autonomous systems (Karim et al. 2004).

Autonomous control provides a potential solution to the issue of human control becoming overly complex or resource intensive (in terms of manpower and equipment), with many scientific, civil and military fields already employing partially or fully autonomous Unmanned Aerial Vehicles (UAV) for search and rescue, disaster recovery and reconnaissance operations (Gu et al. 2000). The ability to define a mission as a series of tasks, which may then be sent to mobile nodes to be sorted, allocated and acted upon is of great benefit when considering ways of increasing the reliability of mobile nodes in the absence of human intervention.

Many mobile platforms, notably quadrotor UAVs, have become popular for use in autonomous MANETs despite significant limitations (Ryan et al. 2004). Low power reserves, payload limitations and limited computational capabilities are critical resource constraints for these platforms. Additional to these limitations are the limitations of the MANET itself, with unpredictable change of network topology, open communication medium and probabilistic delivery of data packets adding complexity to the issue of organising nodes autonomously and completing missions in an optimal manner.

A critical weakness in autonomous MANETs is their security. Without human oversight, such networks are vulnerable to attacks against the wireless medium they use to communicate, as well as the inherently trusting nature of MANETs (Zhou & Haas 1999). Trust is required to allow nodes to cooperate and form serviceable routes (Jawandhiya et al. 2010). It may also be required as a part of any task allocation or cooperation services required to complete a given mission. If a network becomes compromised, either in terms

of its ability to provide network services (routing), or mission-related services (e.g. task allocation, information sharing), then it will fail the assigned mission, with the potential for loss or damage to the nodes. In hazardous scenarios such as disaster management, this loss may even extend to human lives (Phan & Liu 2008).

This thesis documents the research undertaken regarding *Virtual Closed Networks: Optimised Security for Autonomous MANETs*. This Chapter will introduce the thesis, providing an outline of the research that have been undertaken, including research questions, original contributions, and research objectives.

1.3 Research Questions

This research will focus on the vulnerability of wireless MANET communication to attack, seeking to identify methods of securing MANET communication and topology with minimal impact on required services.

The following questions will provide a foundation for investigating the existing requirements of autonomous MANET communication, and how MANET communication may be secured within the constraints identified:

1.3.1 What data, and how much of it, is required to achieve autonomy in a MANET?

Knowledge of the type, amount and size of the data that must be communicated across the network to allow nodes to operate cooperatively is required to analyse the cost of autonomy. The means by which a network allocates tasks and associates nodes with objectives in a larger mission framework must be identified to determine what communication is required to facilitate autonomous function.

Autonomous systems may be localised or distributed, with local system requiring little to no external communication to carry out their assigned function. However, MANET-based systems imply a distributed approach. These use of networking technology to allow

communication between nodes on the network implies that nodes are intended to communicate about the problem at hand, and work together to formulate solutions. These solutions will then be communicated throughout the network and acted upon.

Identifying the network requirements of the communication associated with these vital control processes is a vital element of the research.

1.3.2 How does the need for autonomy drive resource consumption, such as communication complexity and data requirements, in MANETs?

The identification of MANET resource constraints is a parallel inquiry to the one raised in Sub-section 1.2.1. The needs of processes with non-optional communication as a part of their function will consume network resources. The limits of these resources must be identified to determine the maximum level of utilisation that can be supported.

1.3.3 How can a MANET be secured against passive and active attacks?

To explore methods of securing an autonomous MANET, the vulnerabilities associated with the communication medium and method(s) must be identified and investigated. Attacks on networks take two broad forms, passive attacks that rely on observation and information gathering, and active attacks that use the gathered information to perform malicious acts against the target network.

By identifying the particular attacks that may perform these actions against a MANET, and mapping them to the communication required by processes needed to provide autonomous function to the network, a set of specific vulnerabilities may be derived. This informs the proposal of strategies, services and protocols suitable for the identified security needs.

1.3.4 What are the costs associated with providing security?

Having identified the security requirements of a given network, the costs associated with the provision of security may be investigated. Security represents an additional cost, stacking with the resource utilisation of underlying (and now secured) communicating processes.

Identifying these costs allows for the total cost of security under a given framework and protocol suite to be identified. This information allows the assessment of the viability of a given approach to a target network. Over-utilisation of network resources requires that security be made more lightweight, potentially compromising the effectiveness of the proposed solution, or that underlying services be increased in efficiency (or reduced in scope).

1.3.5 How can processes critical to autonomous functionality be made more efficient in terms of use of network resource consumption?

The control processes required to drive an autonomous system are usually verbose, requiring a great deal of communication to provide network-wide solutions. By identifying methods of reducing the overheads associated with this communication, the total cost of control communication may be reduced. This frees network resources for other activities, which may be control or security related.

1.3.6 By what means can the cost of security be reduced, without impairing the provision of security services?

As described in Sub-section 1.2.5, the cost of communication may be reduced by investigating the communication requirements of the underlying process. The same can be done for security, identifying the minimum criteria for effective provision of security under given use-cases. By analysing the cost of security in terms of specific network

services, it is possible to identify solutions which reduce the use of network resources. It is vital that any reduction in network resource use, relative to a comparable framework or protocol, does not compromise the ability of the proposed solution to provide required security services.

1.4 Objectives

The objectives of this thesis are:

- Conduct an investigation of autonomous MANET communication requirements.
- Conduct an investigation of existing approaches to securing such networks.
- Propose modifications to existing problem solving algorithms to reduce communication requirements.
- Propose a novel security framework designed to provide a full suite of security services, as defined in the International Telecommunication Union Telecommunication Standardisation Sector (ITU-T) document Rec X.805, to an autonomous MANET at an acceptable cost in additional communication cost and complexity.
- Design, conduct and document the results of experiments, analysing all proposed approaches to problem solving and security in autonomous MANETs, comparing such proposals against comparable existing solutions.

These objectives are developed further in the Problem Analysis presented in Chapter 3.

1.5 Domain Boundary

The core concepts that provide the foundation for this research are; autonomous function, secure communication, and mobile ad hoc networks. These three areas represent the broad scope of the research, and their definition into research-specific goals defines the boundary of that research.

Autonomous functionality requires that systems remain robust and effective without human intervention for a duration of time. The boundary of the research lies in the analysis of the requirements of task allocation algorithms and similar enablers of autonomy, innovation of novel autonomous services is not considered within the scope of this research. The analysis and potential improvement of the means by which such services communicate between nodes, is the focus of this aspect of the research.

Secure communication is the primary focus of the research. The boundary may be defined as the development of a full security implementation; this research is focused on analysis of proposals and the formation of a theoretical security framework for autonomous MANETs. The implementation of a full security framework is considered to be outside of the scope of this research.

All elements of the research are focused on MANETs. No other network type is considered within the scope of this research, as autonomous MANETs have been selected as the primary focus for the research to be undertaken. The very specific needs of each type of network, be they infrastructural, cellular or otherwise, would create such a degree of variety and complexity that any proposal resulting from the research would fail to address the research topic.

1.6 Original Contributions

This thesis documents a number of novel contributions to knowledge that arise from the work undertaken. The following bullet points summarise the original contributions identified and documented:

- The proposal and analysis of Cluster Formed CBBA, a method of clustering in CBBA to optimise communication and reduce processing complexity.
- The definition of Virtual Closed Networks, a means of providing VPN-like functionality to within a MANETs.
- Full-suite security for autonomous MANETs, in the form of the Security Using Pre-Existing Routing for Mobile Ad hoc Networks (SUPERMAN) framework.

- A vouching system (Security Using Pre-Existing Routing for Mobile Ad hoc Networks (SUPERMAN) referral mechanism) for key exchange to reduce the amount of communication required for multi-hop node authentication.
- Performance analysis of secure routing, comparing Secure Ad hoc On-demand Distance Vector (SAODV), Secure Optimised Link State Routing (SOLSR) and SUPERMAN.
- Performance analysis of security-related control communication, comparing Internet Protocol Security (IPsec) and SUPERMAN.

1.7 Thesis Structure

This document follows the structure outlined below:

- Chapter 2: Literature Review.
    - An extensive body of existing literature is reviewed to provide a foundation of knowledge in the field of autonomous MANET security. This will provide a foundation for further problem analysis and the identification of research gaps that will provide the main avenues of progress throughout the research.
- Chapter 3: Problem Analysis.
    - The research gaps identified in the Literature Review are explored in-depth, with fundamental issues being analysed to define a clear statement of where potential original contribution may be derived. This provides the basis for the proposal of solutions to key research questions, themselves derived from the gaps identified in Chapter 2.
    - Distributed Task Allocation (DTA), specifically consensus based methods of decentralised, are identified as a required vector for investigation.
    - Key security issues arising from the need for autonomy and the architecture of MANETs are identified. The need for a framework to specifically address these issues is identified.
- Chapter 4: Optimised Distributed Task Allocation.

  - Based on the outcomes generated by the Problem Analysis, this chapter proposes two approaches to reducing the communication cost and complexity of Consensus Based Bundle Algorithm (CBBA). These proposals seek to address issues identified relating to the use of multicast and broadcast communication.
  - A cluster-based approach to CBBA is identified. It is hypothesised that the approach will greatly reduce communication complexity, decreasing data and transmission requirements.
  - It is proposed that with the addition of control traffic to provide node synchronisation across the network, broadcast communication will further reduce communication complexity buy cutting down on redundant communication.
- Chapter 5: Results of Optimised DTA Simulation.
  - Presents the results of simulation for the proposed approaches outlined in Chapter 4. This is followed by an analysis of those results, outlining the effectiveness of the proposed solutions relative to existing approaches.
  - Cluster Form CBBA (CF-CBBA) is found to reduce communication complexity and cost significantly. Issues are identified with the optimality of assignments in specific cluster configurations.
  - Broadcast Enabled CF-CBBA (BECF-CBBA) is shown to dramatically reduce communication redundancy by allowing nodes to synchronise without the unicast mechanisms used in baseline CBBA.
- Chapter 6: SUPERMAN: A Closed-MANET Security Framework.
  - Building on the problem analysis conducted and hypotheses put forth in Chapter 3, this chapter proposes Security Using Pre-Existing Routing for MANETs. This is a proposed security framework intended to provide full-suite security to autonomous MANETs, for all services required by such networks.
  - Security Using pre-Existing Routing for MANETs (SUPERMAN) is proposed as a full-suite security framework targeted at autonomous MANETs.

    - The Virtual Closed Network (VCN) approach to MANET security is proposed and detailed, forming the core philosophy of the SUPERMAN framework.
- Chapter 7: Results of SUPERMAN Simulation.
    - Presents the results of simulation for the proposed approaches outlined in Chapter 6. This is followed by an analysis of those results, outlining the effectiveness of the proposed solution relative to existing approaches.
    - The SUPERMAN framework is compared with Internet protocol security (IPsec) and found to generate less security control overhead during both initialisation and security association phases, when forming a secure network.
    - The SUPERMAN framework is found to secure routing more cost effectively that SAODV and SOLSR, with lower data requirements and the preservation of underlying routing protocol functionality without the need for secure behaviour. The VCN approach removes the need for modified protocol behaviour for network services to be secured.
    - The SUPERMAN framework is found to require less security overhead to secure CBBA and CF-CBBA, when compared with IPsec.
- Chapter 8: Conclusion.
    - Closes the thesis, providing an overview of original contributions, how they were achieved and justification of those assertions. Future work is proposed, based on the identified areas in which the research touches on the domain boundary established in Chapter 1.

1.8 Chapter Summary

This chapter has provided an introduction to the thesis. It outlines the purpose of the document, background information, and the original contributions that this research provides. The domain boundary of the research and the structure of the thesis has been defined, to provide the reader with a clear indication of the scope of the research conducted and the format in which it is presented.

1 INTRODUCTION

The next chapter will review existing, published literature relevant to the identification of the current state of the research domain. This will allow the identification of research gaps mapping to the research questions and original contributions.

2 LITERATURE REVIEW

2.1 Chapter Introduction

This chapter will provide a comprehensive review of existing literature related to the research undertaken in this dissertation. MANETs, Distributed Task Allocation (DTA) communication requirements and protocols relating to the provision of network, control and security services in autonomous MANETs are researched.

A body of existing literature will be assembled and investigated throughout the chapter, to illustrate the research domain and identify research gaps. The purpose of this work is to identify research gaps that may allow for the production of original research as a result of pursuing identified open problems.

2.1.1 Chapter layout

The literature review is organised into the following sections:

- Section 2.2 reviews research concerning MANET architecture and issues of autonomy, to provide an overview of capabilities and limitations in such networks.
- Section 2.3 focuses on literature concerning the routing mechanisms of MANETs, to allow analysis of the capabilities of existing routing protocols.
- Section 2.4 reviews literature documenting algorithms for autonomous decision making, to assist in defining the term ‘autonomous MANET’ in the context of this thesis.
- Section 2.5 reports on current literature in the field of MANET security, including routing and communication security.
- Section 2.6 discusses literature related to the field of security frameworks, outlining prominent existing frameworks and proposals.
- Section 2.7 provides the research gap analysis, which identifies the areas of original research that this thesis will focus on.

- Section 2.8 summarises the chapter.

2.2 Mobile Ad hoc Networks (MANETs)

Wireless networks have developed to meet the demand for a fully mobile network topology without supporting infrastructure. This has occurred despite the limitations of the wireless medium such as high error rate, power restrictions, bandwidth constraints and link capacity (Saeed et al. 2012). As nodes may change their position freely, multi-hop routes are required to enable nodes to contact other nodes within the network without needing to be in range of their intended destination node or a base station as in traditional wireless networks. Networks exhibiting the characteristics of free-movement, no supporting infrastructure, and wireless communication are called Mobile Ad hoc Networks (MANETs) (Reidt & Wolthusen 2008).

(Reidt & Wolthusen 2008) identify that MANETs can be sub-divided into types based on the core requirements of their application domain. Aerial, vehicular (VANET) and submarine implementations of MANETs exist, with mobility characteristics dependent on their area of application (Sommer & Dressler 2007).

(Quaritsch et al. 2010) conducted a survey of networked UAVs, identifying that aerial MANETs are most commonly used due to the freedom of movement afforded by their flight capabilities. Lightweight platforms such as quadrotor drones are a typical choice for many applications due to their agility, speed, ability to hover, and three dimensional mobility (Ivancic et al. 2012). A key issue with such platforms are their energy constraints and their power demanding propulsion method, namely propeller powered hover and flight (with no supporting aerofoil to mitigate energy expenditure). This varies between platforms, but agile, lightweight mobile platforms tend towards short operational periods due to power limitations.

The rate of topology change can vary greatly depending on the mobility of nodes in the network. Fast moving nodes may require frequent route updates, as routes that were previously viable become invalid due to the network entering a new configuration unsuitable for the previously suggested route (Enneya et al. 2009). (Comparetto et al. 2003) states that the range of communication may also be affected by the power available

for transmission and the effects of the environment in which the network is deployed. To address the high degree of variability in the communication constraints and mobility of nodes, several approaches have been taken to address the problem of MANET routing.

(Kiran 2009) analyses MANET protocol architecture, identifying the numerous differences between MANETs, wireless and wired networks. Unlike WiFi (IEEE 802.11 b/g/n) and wireline networks, MANETs lack infrastructure, with nodes performing the role of router and end-point as required. WiFi networks share the weaknesses of the wireless medium, but are supported by infrastructure within one hop of end points, allowing many of the security services and protocols available to wireline networks to be applied reliably (Bakshi et al. 1997). Such is not the case for MANETs, which must involve all nodes in network-wide services.

2.2.1 Autonomous MANETs

A key issue identified in the effectiveness of MANET routing protocols is the manner in which they respond to changes in network topology. The speed with which new routes may be planned and the complexity of the communication required to plan those routes are limiting factors when considering the division of network resources between routing and other required services.

(Bellur et al. 2002) identify a key issue in autonomous MANETs in that they must maintain frequent communication with one another to avoid collision, repetition of labour (performing the same task when only one node needs to service it) and provide routes for communication between distant nodes. This extends to redundant activity, nodes must communicate with all other nodes involved in a task allocation process to avoid repetition of tasks undertaken by their peers (Vincent et al. 2003). In large networks, this may represent a considerable amount of communication.

(Bethke et al. 2008) expanded on the topic of communication load in MANETs, taking into account the health of autonomous system. Health is defined as the combination of all variables related to the operational lifespan of an autonomous system, such as UAVs. This attribute can include varying traits of a platform, network performance, energy

reserves and platform specific payload performance. The purpose of monitoring the health of an autonomous system is to determine the fitness of a node for continued use in a mission. Specific to autonomous MANETs is the concept of network performance. Should a node become unreliable, failing to communicate for a period of time or suffering significant loss or corruption of transmitted packets, it may be deemed unfit for duty and retired from the mission. An important observation made by Bethke et al., is that the health of a system may be adversely affected by choices made in the type of networking technology used.

The network must therefore provide reliable routing to ensure that nodes may communicate in a timely manner, with minimum loss of communicated data. The high mobility of nodes in some scenarios can complicate this, reducing the reliability of the MANET routing protocol depending on the type of protocol chosen.

In recent work, (Gundry et al. 2012) identify that autonomous control and MANET topology is closely linked. It has been previously identified that reliable routing is required to allow the communication of control data, but the distribution of nodes in a network may also have effects on latency and quality of service.

(Kusyk et al. 2013) reinforces these observations, conducting a quantitative review of an autonomous MANET performing a mapping mission. It is proposed that the coverage of a network will play a large role in the optimality of mission involving a high-level of mobility. By ensuring that nodes form a network that provides maximum connectivity while reducing individual movement in the mission-space, energy may be conserved and MANET performance improved. Balancing the mobility of nodes with the communication requirements of task allocation and control is therefore a vital consideration when developing autonomous systems for the MANET paradigm.

2.2.2 Section Summary

MANETs can be considered to be a new type of network, a proposal made in many of the publications cited. MANETs allow nodes complete freedom from the confines of

traditional network infrastructure, at the cost of requiring each node to participate in routing and network-based services.

The highly mobile nature of some MANETs means that they have the potential to experience rapid changes in their topology. The mobility of such networks allows them to provide useful services in a variety of civil applications. This desirability is offset by the complexity of MANET architecture, specifically the communication complexity of ensuring reliable routes between nodes.

MANETs are a popular emerging network architecture that has proven suitable for a wide variety of applications depending on highly mobile, independent nodes. (Gundry et al. 2012) state that limitations come in two forms:

- Limitations of architecture (and the wireless medium).
- Limitations of platforms used as MANET nodes.

The limitations of MANET architecture centre on the wireless medium and the organisation of communication in the absence of specialist routing infrastructure. The wireless medium is open, in the sense that outside observers may freely capture and analyse data packets transmitted by wireless nodes. Without routing infrastructure, each node must play a role in topology generation. As a result, the previously discussed wireless medium problems may also affect communication required by the routing protocol.

Platform limitations stem partially from the carrying capacity of the target platform. Lightweight UAVs, for example, will not be able to carry heavy-weight hardware or power supplies. This limits their individual utility, as they may require specialist payloads and have short operational lifespans before needing to have their batteries replaced. The cooperation requirements of specialised UAVs may also increase the complexity of communication and task allocation, as nodes may need to work together to combine their limited unique capabilities to address problems unsolvable by individual nodes.

2.3 MANET routing protocols

The management of a dynamic topology with many participating nodes is non-trivial, routes may become broken due to node movement or changes in local conditions that affect the communication medium (Lee et al. 1999). MANET routing protocols have been proposed to address this problem and provide reliable routing in changeable networks. Three key approaches have been adopted: reactive, proactive and hybrid (Royer & Toh 1999). These approaches use routing tables to store route information locally on each node. Due to the infrastructure-less nature of MANETs, every node is considered to be a router, in addition to any other role they might perform in the network.

Table 2-1 shows the key attributes of proactive and reactive protocols, which will be discussed with reference to the reviewed literature in the relevant sub-sections. Hybrid routing protocols are not identified, as they vary in which attributes of proactive and reactive routing they utilise.

Table 2-1 Table to show the attributes of Proactive and Reactive MANET routing protocols (Chandra 2005)

	Proactive (Table-driven)	**Reactive (On-demand)**
Availability of Routing Information	Immediately from route table	After a route discovery
Route Updates	Periodic advertisements	When requested
Routing Overhead	Proportional to the size of the network regardless of network traffic	Proportional to the number of communicating nodes and increased with increased node mobility

(Daranasi et al. 2012) identify node mobility as a key consideration when selecting an appropriate routing protocol. By performing a quantitative analysis of three routing protocols and three mobility models using Network Simulator 2 (NS2), Daranasi et al identify that AODV, DSR and DSDV respond differently to Manhattan Grid, Gauss Markov and random waypoint mobility models, each exhibiting differing packet delivery in each scenario. Observations made by (Maan & Mazhar 2011) demonstrate that

proactive routing protocols are also affected by node mobility, much in the same way as their reactive counterparts. The movement of nodes can therefore be said to have a variable effect on the reliability and quality of service of a MANET, and fore-knowledge of likely patterns of motion in the field can inform the selection of appropriate routing protocols.

2.3.1 Reactive MANET routing protocols

Reactive routing protocols provide on-demand route formation services. This means that routes are discovered when necessary, routes are not maintained between all nodes. Routes are maintained only as long as is necessary, as determined by a lifetime value set within the chosen protocol.

Figure 2-1 illustrates the communication involved in planning a route between two nodes over multiple hops using AODV.

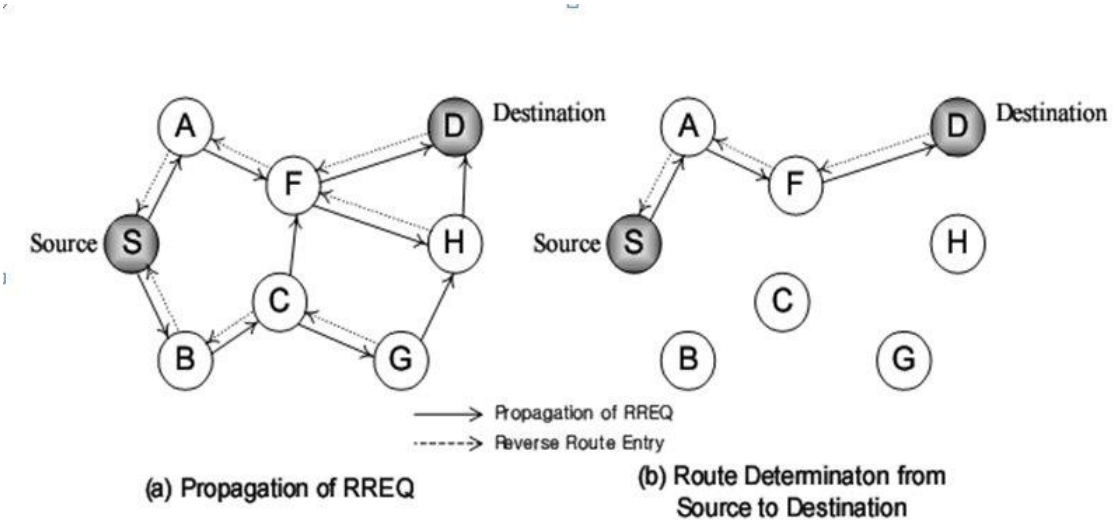

Figure 2-1 Diagram showing the generation of a route between nodes S and D using AODV [http://flyingdcat4.tistory.com/entry/AODV]

(Perkins & Royer 1999) propose Ad hoc On-demand Distance Vector (AODV) protocol as an example of a reactive routing protocol, which is now an IEEE standard (RFC 3561). AODV provides the indicated attributes, while also providing lifetime tracking of routes,

unicast and multicast routes, and sequence numbering (Chakeres & Belding-Royer 2004). These services prevent underutilisation of routes, the formation of redundant routes and looping back over nodes that have already participated in route planning respectively. These services are vital to ensure that routes are created quickly and efficiently.

Reactive routing protocols tend to require comparatively complex communication for route planning. AODV requires a three-way exchange between nodes along a route to plan and verify that a route has been formed between source and destination nodes. In networks with a slow rate of topology change, the lifetime of routes that are being utilised may be long, and therefore the complexity of routing is mitigated by the duration for which those routes persist (Lee et al. 1999). In highly mobile networks, the rate of change in topology may be rapid enough that routes frequently break. Under such circumstances, reactive protocols may become network-resource intensive (Kuppusamy et al. 2011).

Recent work regarding reactive routing protocols includes a diverse array of topics; tactical communications and energy-saving are prominent examples. (Cheng & Moore 2012) evaluate the performance of AODV against that of Optimised Link State Routing (OLSR) and Open Shortest Path First MANET Designated Routers (OSPF-MDR) for tactical communications. Overhead traffic and end-to-end delay are the primary variables observed; with AODV being found to provide low-overhead in low mobility scenarios when compared to OLSR. The cost of re-routing is cited as the primary driver of cost under AODV; the previously identified cost of forming new routes when existing routes become invalid or time-out is reinforced in this contemporary analysis.

(Bade et al. 2013) investigate the effective use of energy saving measures under AODV. The cost of routing under reactive protocols is identified as a critical issue if frequent re-planning of routes is required. Mitigation of cost through constrained mobility is not always a viable option. Bade et al propose an algorithm to alert autonomous systems of energy thresholds, affecting their behaviour based on energy consumption, extending the lifespan of nodes by manipulating node topology based on node energy demands.

2.3.2 Proactive MANET routing protocols

Proactive routing protocols take a different approach to reactive protocols, by generating routes between nodes periodically. Instead of requiring that a node identifies a destination node and requests a route be formed between itself and its target, routes are planned between all nodes in advance (Jacquet et al. 2001). This is performed on a regular basis, with nodes polling each other for neighbour information that may be used to gain partial or full knowledge of the network topology, depending on the protocol used.

(Clausen. & Philippe 2003) propose OLSR as a protocol designed with this philosophy in mind. OLSR nodes use HELLO messages to discover route information for nodes within two hops whenever a network poll is performed (Clausen. & Philippe 2003). OLSR differs from many proactive routing protocols by making use of Multi-Point Relays (MPRs), elected nodes which have a high number of two hop connections. When planning a route between a source and destination, these relays are used as critical points; not all nodes are polled regarding the existence of a given route, only MPRs. This reduces the amount of communication required to plan a route between two or more nodes.

A criticism of the proactive method is that it requires a synchronised topology database across the network (Kuppusamy et al. 2011). MANETs cannot guarantee synchronisation of nodes, making the development of an appropriate algorithm difficult. This issue is mitigated by polling frequently enough that even if some polls are not received by all nodes, subsequent polls allow the inclusion of those nodes in the routing process. These polls are performed by flooding topology data throughout the network, with the frequency of flooding set at a rate that ensures the network does not become unsynchronised for a significant period of time. Figure 2-2 shows the two phases of routing under OLSR: the initial polling phase, where nodes flood routing packets to generate partial topology information, and the selection of a route by Node 4, with the dark shaded nodes (MPR nodes) relaying messages to the intended destination (Node 9) by the shortest path.

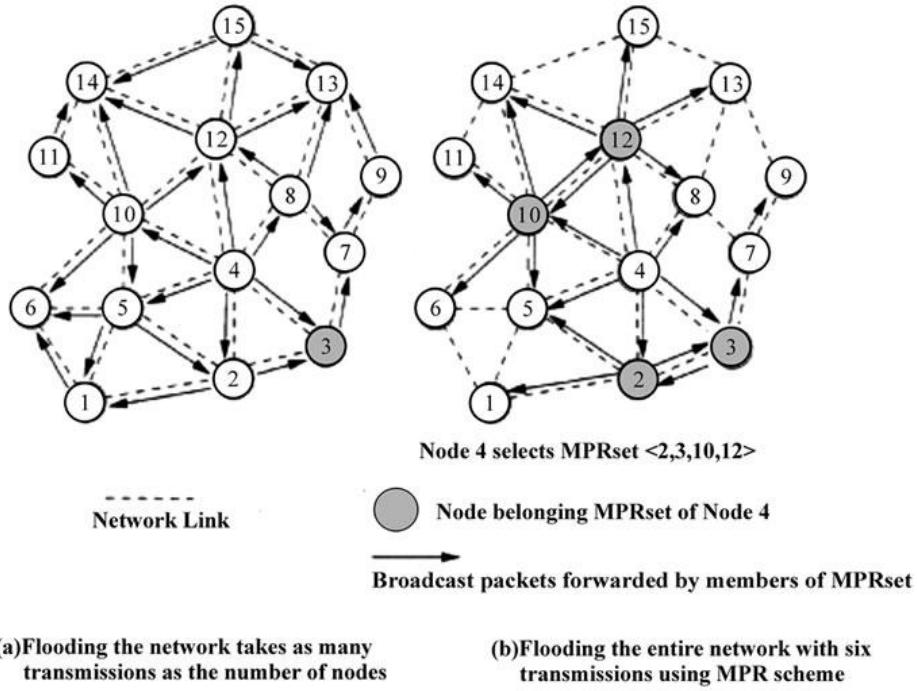

Figure 2-2 Diagram showing the flooding and routing mechanism of OLSR [Enneya et al. (2009)]

The flooding mechanism may appear to be communication intensive, but packet sizes are kept small to reduce the impact this has on network resources (Guo et al. 2010). For low-mobility scenarios where topology change occurs slowly, reactive protocols may provide a more efficient mechanism for routing in terms of utilisation of network resources. (Hinds et al. 2013) provides a contemporary review of routing protocols for MANETs. Proactive protocols are identified as suitable for high mobility nodes, due to their periodic-poll routing mechanisms. This reinforces the observations of Guo et al that the selection of a suitable routing protocol is partially dependent on node mobility characteristics.

Proactive protocols are highly responsive to changes in topology, due to their frequent polling mechanism. In situations that involve highly mobile nodes, proactive routing protocols may generate routes as quickly as the network changes, giving them an advantage over reactive protocols in terms of routing latency and network resource utilisation.

2.3.3 Hybrid MANET routing protocols

Hybrid protocols such as the Zone Routing Protocol (ZRP) attempt to combine the positive attributes of proactive and reactive protocols (Haas et al. 2002). Partial knowledge of the topology is generated proactively, with a reactive component allowing additional routes to be formed on-demand with nodes that are not discovered as part of the initial route prospecting service (Chahidi & Ezzati 2012).

The number of nodes found during the discovery phase has an impact on the effectiveness on the proactive element of such protocols. If a low number of routes is discovered this way, more demand will be placed on the reactive element. Similarly, the reactive component of the protocol has the same disadvantages as purely reactive protocols in highly mobile networks experiencing high communication loads.

Rapid changes in the attributes used to qualify which nodes are reached proactively, such as signal strength, adjacency and hop count, can cause the responsiveness and resource utilisation of hybrid protocols to vary significantly (Jayakumar & Gopinath 2007). This makes it difficult to predict how hybrid protocols will perform in high mobility scenarios, as the rate of topology change will have a variable effect based on how many nodes qualify as being local to others during the proactive flooding stage, which in turn effects how frequently the reactive component of the protocol will be invoked to generate additional routes.

2.3.4 Section Summary

MANET routing protocols define the manner in which network topology is generated and maintained. A relationship has been identified between the manner in which nodes move with respect to one another, and the complexity of routing under reactive, proactive and hybrid MANET routing protocols.

This research establishes the fundamental aspects of MANET routing, identifying investigations and analyses of the requirements and constraints of the three key types of MANET routing protocol.

2.4 Distributed Task Allocation (DTA)

Distributed Task Allocation (DTA) algorithms are a class of algorithms that concern the distribution of a set of tasks between a set of nodes. DTAs are used within autonomous MANETs for the division of work among nodes. Such algorithms are described as centralised or decentralised.

Centralised DTAs nominate a single node as the controller for the task allocation operation. This node requires that other nodes provide it with their state information relevant to decision making, and is required to communicate the results of the task allocation back to those nodes. Centralised approaches have been shown to require significant communication between a single decision making node and the rest of the network. Such approaches also require that the central controller node have the processing capabilities to generate solutions for the whole problem, with no sub-division of labour (Botelho & Alami 1999). They do not require communication whilst undertaking the computation, which provides benefits in terms of network resource utilisation. Centralised approaches suffer from having a single point of failure. If communication is lost with the central controller, the network loses all decision making capabilities. Non-controller nodes may not have the hardware or software to perform DTA. Even assuming that all nodes are capable of DTA, a new controller must be nominated, which is a potentially time consuming election process requiring all nodes to communicate and nominate the new controller before any further DTA processes can be started.

Decentralised approaches require partial or full network participation in the decision making process (Jin et al. 2003). Instead of relying on a single node to distribute a solution, each node in the decision making set will provide a partial solution to all other nodes in their set. This requires communication between all decision making nodes to reach a solution, which is then communicated to all nodes that did not participate in decision making.

Partially decentralised DTA requires that a set of decision making nodes be nominated. The use of multiple decision making nodes reduces the impact of the loss of DTA capable nodes, as the remaining members of the set may continue to perform DTA (Saad et al. 2011). Communication is more complex under such systems, as all members of the

decision making set must share information and agree on a final allocation that is deemed optimal for the network. The decision making nodes must also agree on their responsibilities for processing state information from specific nodes outside of that set. This approach reduces computational complexity by requiring that each node only makes decisions based on their assigned non-set peers, and comparison of their solutions with decision making set peers. This comes at the cost of additional communication complexity.

Fully distributed methods take this a step further, requiring that all nodes participate in DTA (Ostergaard et al. 2001). Each node bids for itself, and all other nodes provide counter bids where their suitability for a task exceeds that of bids placed by their peers. Nodes are responsible for the computation of their local DTA solution and the communication of their solution to all other nodes in the network. This further reduces the computational complexity of DTA on individual nodes, at the cost of even higher communication complexity than partially decentralised methods.

The manner in which nodes agree on a solution is determined by the approach taken; market-based auction or consensus-based DTA.

2.4.1 Market-based Auction Approaches to DTA

(Zlot & Stentz 2006) propose a market-based auction model, inspired by the distribution of resources through economic principles. In their model, tasks are given a cost, which is a variable that is an abstraction of the resource consumption likely to be incurred by performing the task. This provides the benefit of node by node calculation of the cost of a task, distance from an objective, method of propulsion and other factors may be integrated into the cost equation allowing for individual assessment of a task's value. A second variable, referred to as a score, represents the reward element of completing a task. This is an arbitrary value assigned by the designer of a set of tasks, to represent the importance of one task over another.

Nodes bid on tasks, communicating their fitness to perform a task to either a central controller (in centralised implementations) or all other nodes in the network (in

decentralised implementations) (Gerkey & Mataric 2003). By subtracting the cost of a task from the score, a node is able to determine whether it is worth bidding for that task. By comparing the bids of other nodes, the most suitable node for a task may be determined.

(Ducatelle et al. 2009) identify that centralised implementations of market-based models require that nodes communicate either their bids, or their current position and state information to a central controller. This controller then processes bids to create tasks lists, which are then communicated back to all other nodes in the network. Only relevant information is communicated, so nodes are only made aware of the tasks for which they qualify.

Decentralised methods require round-robin communication; either between a set of nodes identified as decision makers that bid on behalf of associated nodes, or all nodes participating equally. (Drucker et al. 2012) demonstrate that the amount of communication during the decision making process will be far greater than that of centralised methods. There is no single point of failure in a decentralised method, meaning that the loss of a node only impacts the resources the network may assign to solving a problem.

(Johnson et al. 2012) address a critical issue in purely score-based market approaches; scores given may provide sub-optimal ordering. Nodes may be sent to points in the mission area that bypass tasks lower in their task-list. Non-submodular scoring provides a means of ordering tasks in a more efficient manner, but as identified in sub-section 2.2.3, the communication required to avoid colliding with other nodes may increase as the path that a node is following may differ from the task information available to other nodes. As a result the order in which tasks are undertaken may be made more optimal, but communication requirements may be increased.

A key constraint with auction algorithms is identified by (Dasgupta 2012) as their inherently sequential nature. Tasks are auctioned one at a time, which results in a large amount of wasted communication for nodes which do not win a bid. Although the participation of all nodes in the bidding process is required to select the optimal bid, those that do not receive the task do not gain anything from that specific exchange. This has resulted in many centralised approaches to offset the perceived communication cost, with

the previously discussed issue of single-point of failure and communication load on the central controller that scales with the size of the network.

Risk management is considered in contemporary literature. (Ponda, Johnson & How 2012) state that a risk of centralised and partially distributed systems is sub-optimal task allocation at the local level. This is a result of purely market-based models repurposed to provide globally optimal solutions, as the burden of sub-optimality is passed to the individual nodes, rather than the network as a whole. The need for solutions that focus on globally optimal distributed task allocation is noted, with reference to existing algorithms.

2.4.2 Consensus-based Approaches to DTA

(Ren & Beard 2003) introduce the concept of consensus seeking in distributed task allocation problems, investigating the issue of multi-robot control. Purely market-based models tend towards single-robot single-task allocation, where one task is given to one robot and the network uses information from all nodes to determine the recipient of the task. Consensus-based methods seek to reduce wasted communication by allowing multiple robots and multiple tasks to be processed simultaneously.

Consensus-based task allocation uses auction or market-based allocation as a foundation, following the same principles of bidding, scoring and perceived cost. In single task implementations, such algorithms required the agreement of all nodes that a given node is the most suited to a task, but this has the same issue of inherently sequential processing discussed in sub-section 2.3.1.

(Brunet et al. 2008) propose Consensus-Based Bundle Algorithm (CBBA), an algorithm that allows nodes to submit a collection of tasks for review by their peers, instead of only handling one task at a time. This allows nodes to perform calculations on the whole problem domain, determining which they are able to bid on and submitting the results of those bids to all other nodes in the network. By performing rounds of bidding and counter-bidding, the network will arrive at a state where no modifications are made to bundles in a round. This state is referred to as consensus, and the resulting bundles are then committed as task allocations to the participating nodes.

The allocation mechanism can be further refined through the use of non-submodular scoring, and task restriction based on node capabilities (Hunt et al. 2012). It must be stated that refining the allocation of tasks does not address communication issues. Although CBBA and equivalent algorithms mitigate wasted communication by allowing nodes to communicate a large number of tasks for processing in one round, communication is still inherently sequential.

CBBA uses round-robin communication to allow all nodes to transmit their bundles to all other nodes. This communication can be assumed to be sequential due to the need for coherent state information at each stage of the DTA process on each node. This leads to a steep rise in communication complexity with each node added to the network.

Early implementations assumed unicast communication, to address the issues encountered when networks were not arranged with all nodes as immediate neighbours (Johnson et al. 2010). (Johnson et al. 2011) proposed Asynchronous Consensus Based Bundle Algorithm (ACBBA) to address the asynchronous channel problem, improving the efficiency of communication by allowing multicast communication and an event-driven initialisation of bundle sharing instead of the synchronised sequential method formerly used. Full utilisation of the broadcast capabilities of wireless communication has thus far been avoided due to the communication cost associated with control of network flooding mechanisms as a means of state information propagation (Ponda, Johnson, Kopeikin, Choi & How 2012).

Large networks will encounter significant communication load, with significant wait times between the start of a round and its completion, as identified by (Argyle et al. 2011). In time limited missions, or on energy constrained platforms, waiting for solutions for long periods of time may not be feasible.

Figure 3 shows how tasks are allocated to nodes using CBBA. The task allocation planner element is present on all participating nodes; for every resulting task list, a node has processed its bids and transmitted a bundle. This reinforces the previous observations made by Hunt and Johnson regarding the inherently sequential nature of CBBA, even when task allocation itself is performed on groups of tasks.

DTA algorithms, including CBBA, tend towards intolerance of network disruption (Olfati-Saber & Murray 2004). In the case of CBBA, partial rounds are unacceptable, requiring a retry of the entire round (Choi et al. 2009). When considering the critical requirement of network stability, the nature of wireless communication may itself cause issues. Implementations seeking to mitigate this issue have been proposed, but unreliable communication may add a significant delay to the completion of a DTA process (Pi 2011). As previously discussed, the time taken to allocate tasks and the use of network resources during that time are critical factors when considering the constraints of DTA applied to autonomous MANETs.

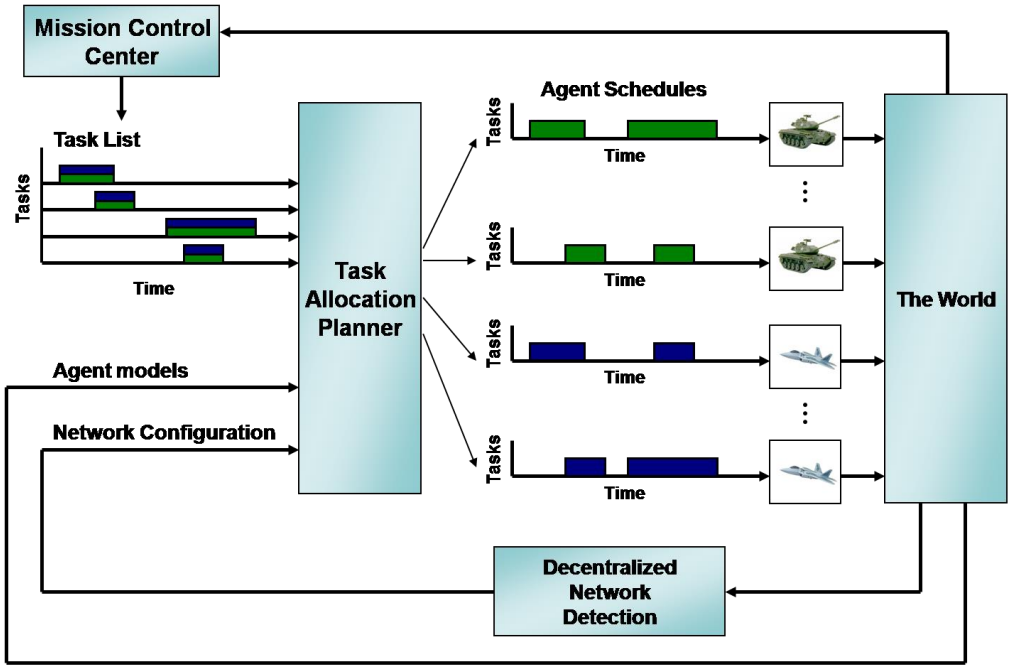

Figure 2-3 Diagram showing the allocation of tasks to ground-based and aerial nodes using CBBA [Choi et al. (2009)]

Contemporary work by (Colistra et al. 2014) maps the problem of task allocation in the context of the consensus-based approach to the Internet of Things. Massively distributed networks with very high numbers of nodes are identified as desirable and an imminent area of focus for consumer goods. The management of task allocation in such networks is a problem, due to the costs associated with routing and the utilisation of the communication channel between related nodes.

The extension of these considerations to logistical networks, such as those used in shipping and commercial distribution, highlights an additional area of application. (Kim & Choi 2014) identify that not only is the number of nodes involved in task allocation important, but the dynamic mobility of those nodes must be considered. The allocation of tasks in networks of considerable size (many thousands of nodes) with highly variable mobility is being addressed by proposed algorithms such as Consensus Based Auction Algorithm (CBAA) and CBBA (alongside application-specific derivatives), but remains an open problem at present.

2.4.3 Section Summary

DTA is required to allow autonomous task allocation. It allows nodes to share their state information relevant to the problem at hand and bid on tasks that represent simple elements of the whole problem. However, many approaches require a highly reliable network with significant computation and communication resources.

The sequential nature of assignments in non-bundle algorithms make them ill-suited to networks with limited network resources, as there is a high degree of wasted communication due to only one node being able to be allocated a given task in each round. CBBA solves the problem of being limited to single task allocations, but still suffers from being inherently sequential and reliant on round-robin communication between all nodes.

The mobile nature of nodes and inherently probabilistic nature of wireless communications encourages the use of decentralised methods due to the communication and computational costs associated with losing contact with the controlling node in centralised approaches. However, decentralising the DTA process increases communication complexity. This complexity increases exponentially as the network grows in size, making current methods unsuited to large scale deployment on MANETs.

The underutilisation of wireless communication capabilities is common in currently proposed DTA algorithms. CBBA and ACBBA both require that nodes send their bundles in a sequential or event driven manner, leading to repetitive communication and a lack of parallelism in the information sharing stage of an allocation round. This represents an

open problem, as the broadcast nature of the wireless medium may be used to allow for broadcast or multicast communication of task bundles to the rest of the network, reducing duplicate communication.

DTA algorithms represent the means by which an autonomous MANET may operate independently of human control. Current approaches provide a means by which small networks of autonomous nodes may share information and arrive at optimal solutions for a variety of simple tasks. Investigation into the means by which communication complexity and time taken to arrive at a solution in large networks may be reduced is identified as a key point of interest.

2.5 MANET Security

Wireless networks suffer from utilising an insecure medium. Wireless transmissions are inherently broadcast, this means that communication may be intercepted as long as the intercepting node is within transmission range.

MANETs rely on routing to determine the topology of the network and ensure the provision of network-related services to all nodes. Disruption, subversion or denial of routing services will have a significant impact on the ability of the network to deliver those services. It is therefore critical that routing services are secured, to guarantee the validity of routes within the network.

As discussed in sub-section 1.3, autonomous networks require control communication to derive solutions to problems and allocate tasks appropriately. Theft, manipulation or destruction of data related to task allocation can have significant consequences. Nodes may be misinformed of which tasks they are expected to perform, or a form of Denial of Service (DoS) attack may be caused by destroying vital control data in transit.

2.5.1 Security Threats

(McGee et al. 2004) investigate the use of ITU-T X.805 as a means of planning secure networks. They identify security and reliability as the key concerns of service providers and enterprise networks, with an emphasis on cost effectiveness. The potential for high-value loss (in terms of finance, reputation or possibly the risk of hazard to human well-being as a result of compromise) should a network be compromised for even a short amount of time is deemed to be unacceptable. These concepts are equally applicable to the domain of autonomous MANETs, as the costs (in terms of time and resources) of having an insecure network are high. Debilitation or outright loss of network and control services is unacceptable, especially as autonomous MANETs may be beyond the reach of potential human intervention.

Table 2-2 Table of Security Dimensions mapped to Security Threats (Richard et al. 2010)

Security Dimensions	**Security Threats (effects on data/information)**				
	Destruction	Corruption Modification	Theft Loss Removal	Disclosure	Interruption of services
Access control	☑	☑	☑	☑	
Authentication			☑	☑	
Non-repudiation	☑	☑	☑	☑	☑
Data confidentiality			☑	☑	
Communication security			☑	☑	
Data integrity	☑	☑			
Availability	☑				☑
Privacy				☑	

The X.805 document details a security architecture that separates networks into three planes; client, control and network. The client plane includes services extended to users, the control plane details any administrative or process control services required by

autonomous elements of the network and the network plane is described as all topology generation and maintenance processes. Table 2-2 details the five key threats to a network and the eight dimensions of security that may be applied to mitigate them.

(Richard et al. 2010) evaluate the IEEE 802.15.4 standard using X.805 as a benchmark for security standards. They find that this framework adequately described the threats that wireless networks face, stressing the need for a full security solution for wireless communications that do not make use of infrastructure. It is also stated that low-power wireless networks have severe limitations on the bandwidth that they can spare for security services, although this is a technological constraint rather than an issue with the medium itself. The evaluation provided is application specific and only concerns itself with end-to-end security. This is an issue when considering the multi-hop nature of MANETs.

It cannot be guaranteed that a network will maintain direct connections between all nodes, due to the probabilistic nature of wireless communication leading to the potential for periodic disconnects when link quality is poor. This may lead to issues in situations where an intermediate node on a route has been subverted, allowing for a variety of attacks. Point-to-point security must also be considered when nodes themselves form the basis of network topology maintenance.

(Chandra 2005) classify MANET security threats as active and passive. Active threats include denial of service, jamming, masquerade, fabrication and modification. All of these threats require the active participation of malicious nodes within the network. To launch active attacks, such nodes must usually engage in passive attacks to gather information about the target network. Passive threats include data packet and traffic analysis. Such attacks may allow security information to be derived, for example any released plain text may allow attacks on cryptographic systems. They conclude that the mitigation of passive threats decreases the likelihood and effectiveness of active threats.

(Garg & Mahapatra 2009) identify security threats specific to MANETs. The key weaknesses of MANETs are reported to be:

- Dynamic topology has been discussed previously, but in terms of security, unpredictable changes in network topology may allow for the injection of

malicious nodes. The directionality of links may also vary, presenting multiple avenues through which to perform session-oriented attacks.

- Bandwidth constraints put a hard limit on the complexity of security that may be applied. The complexity of cryptographic algorithms and the size of packet security fields are two examples of bandwidth consumption by security services. Complex key exchange mechanisms may also run into issues with bandwidth constraints.
- Energy limitations are most noticeable in MANETs designed for energy optimisation. As the energy capacity of mobile nodes are finite, attacks that prevent communication or promote wastage of resources are particularly effective.
- Limited physical security concerns the safety of the device itself. It is possible that a mobile node may be stolen, allowing an attacker direct access to security information and other relevant data. The implementation of persistent network wide keys is therefore inadvisable.

(Bhatia & Shah 2013) provide a contemporary review of security approaches to vehicular ad hoc networks (VANETs). These are essentially long-lived MANET nodes with a wide range of motion, possibly constrained by transport infrastructure. The need for novel security approaches in resource limited networks is identified, due to the proliferation of MANETs as a means for communication exceeding the provision of standardised, resource-aware and efficient security systems.

2.5.2 Secure Routing

(Burmester & de Medeiros 2009) discuss the issue of security related to route discovery. They stress the importance of route discovery as the foundation element of network topology maintenance in MANETs. Securing route discovery is identified as a critical consideration when discussing MANET security, as any manipulation or disruption of routing services will affect the entire network.

(Deng et al. 2002) analyse the security threats that afflict MANET routing protocols, identifying the black hole attack as an attack of particular note. The open nature of the wireless medium and its reliant on cooperation between nodes are identified as key weaknesses to a variety of attacks. The black hole attack sinks routing packets, preventing them from being used to generate routes between nodes. A method to address the black hole attack is proposed for AODV, but does not address other attacks.

(Zapata 2002) proposes Secure Ad hoc On-demand Distance Vector (SOADV) protocol. SAODV extends AODV, providing security services to the routing process. Two mechanisms are used: digital signatures to ensure the integrity of non-mutable fields in routing packets, and hash chains to secure hop count information, which changes with each hop. Route error messages are protected by digital signatures, which may only be propagated over a single hop due to the large amount of mutable information they contain. The purpose of this extension is to provide secure routing; only services related to MANET routing are protected. No additional protection is extended to data sent over a route once it has been established.

(Papadimitratos & Haas 2003) state that just because a route is secure and up to date, it cannot be assumed that the route remains secure. They go on to posit that an intelligent attacker can pretend to behave in accordance with the secure protocol, while injecting malicious packets into the network. This reinforces the previous observation that secure routing does not extend its services beyond the network plane.

(Hafslund et al. 2004) propose a secure extension to OLSR, using digital signatures to provide guarantees of routing packet integrity. Timestamps are also used to prevent packet replay attacks. They note that the addition of digital signatures places additional overhead on the routing packets, an observation that is alluded to in the previously discussed publications, but not explicitly identified as a key feature.

A common feature of both proactive and reactive secure routing protocols is a focus on integrity over confidentiality. Digital signatures are used to ensure that non-mutable field have not been tampered with, but the previously discussed protocols do not encrypt packets. The rationale for this is that information about the network topology may be derived by collecting route requests and observing the addresses, in the case of reactive protocols.

Trust-mechanisms are another method of providing security services. Generally, trust-based protocols give a numeric value to every node in the network that represents a level of trust. Misbehaviour causes neighbouring nodes to reduce this value. If a threshold value is reached, nodes inform other members of the network of the misbehaving node and cease communication with the offending node.

(Thanigaivel et al. 2012) proposed Trust based Routing Mechanism Using Non-cooperative Movement in MANETs (TRUNCMAN). The purpose of this protocol is to isolate and cease cooperating with nodes that abuse the implicit trust inherent in routing operations. This is a purely trust-based system, which does not provide integrity or confidentiality services explicitly. TRUNCMAN can extend a measure of protection to data packets by detecting misbehaviour in the use of routes, but like SAODV and SOLSR, it does not specifically address the security of packets that do not belong to the network plane.

Trust mechanisms allow the identification of misbehaving nodes. (Dhanalakshmi & Rajaram 2008) propose that the detection and isolation of misbehaving nodes will prevent their interaction with legitimate nodes, extending protection to the uncompromised elements of the network. It is possible that by identifying and then blacklisting such nodes, the network may protect itself against attacks that use the identities of formerly legitimate nodes to further compromise the network (Sen 2010).

Risk-awareness is a prominent topic in recent literature. (Zhao et al. 2012) analyses the issue of risk-awareness in MANETs. The risk of attack may be used as a measure of how likely an attack will be, allowing a balance to be stuck between the cost of security and the needs of other network services in resource constrained networks. This highlights the need for flexibility when resource constraints demand a choice be made between a certain degree of protection and the availability of resources for critical network services.

2.5.3 Data Security

Routing ensures that nodes may communicate with one another, over intermediate nodes should it be required. Communication itself is a complex element of any network,

representing not only the data sent over the network, but the manner in which it is sent. The control and client planes of a network require reliable communication to provide services to the network and allow nodes to participate in cooperative processes such as DTA.

(Yang et al. 2004) analyse the security requirements of MANETs, focusing on routing but discussing issues related to data security. The implementation of existing security services is discussed, concluding that the direct application of services developed for wireline services frequently exceeds the limited bandwidth available to many MANETs. This is an observation made in many of the previously cited publications that address MANET security.

Confidentiality services are provided by cryptography, obfuscating the contents of a packet so that if it is intercepted, it is not immediately readable. Secure MANET routing is generally assumed to use pre-shared keys to digitally sign packets for the purpose of integrity, not confidentiality. However, such keys may also be used for cryptographic purposes. It is not safe to assume that keys are pre-shared, as (Liu et al. 2013) state in their paper regarding a framework for distributed key management in MANETs.

(Papadimitratos & Haas 2006) propose an end-to-end security protocol that provides confidentiality and integrity services to data transmissions in MANETs. By providing security between a source and destination, the inherently unreliable and insecure nature of MANET routes can be mitigated. This is achieved by using pairwise keys between nodes to encrypt the contents of packets and sign packets to provide confidentiality and integrity respectively. The approach taken in this publication does not account for the integrity of the route, though it is proposed that the services discussed be applied alongside the routing security protocol proposed by (Papadimitratos & Haas 2003). Even assuming the integration of such security services, point-to-point security is not observed. It is possible that secure packets are intercepted or rerouted, as there is no security applied to the packet between intermediately nodes.

(Liu et al. 2013) stress the need for lightweight, reliable, flexible security that is aimed specifically at the needs of MANETs. Adopting methods that have proven effective on wireline networks may not be possible in many cases due to bandwidth constraints, and may not address vulnerabilities specific to MANETs. Elliptic curve cryptography is

suggested as a lightweight method of forming pairwise keys. Diffie-Hellman key exchange algorithm is also suggested as a viable means of sharing keys between nodes and forming pairwise secure links.

(Maity & Ghosh 2012) focus on the open medium issue of wireless network, suggesting that the solution is to close access to the network. The mobility of nodes in a MANET prevents the use of traditional means of closing a network, such as a firewall. The network must be closed at the node level, with authentication policies enforced to prevent the entry of malicious nodes to the network. Achieving the degree of network closure possible in wireline networks is stated as being beyond the reasonable scope of a closed MANET.

2.6 Security Frameworks

2.6.1 Internet Protocol Security (IPsec)

The IPsec working group created the IPsec security framework, a collection of protocols for wireline networks that provides end-to-end security services. As an open standard, IPsec provides a solid foundation for further work regarding end-to-end communication, as stated by (Doraswamy & Harkins 2003). IPsec has been used to secure internet, intranet and virtual private network (VPN) communication. Unlike the previously discussed secure routing protocols (e.g. SAODV, SOLSR), IPsec seeks to secure data and does not take into account routes. It forms end-to-end security associations that ignore the role of routing in the delivery of packets. This is because of the highly changeable nature of the connection between end-points and the dependable nature of the medium (wired or wireless). The medium and supporting infrastructure provide delivery guarantees, so IPsec only has to account for the integrity and confidentiality of end-to-end communication.

The rationale for using IPsec is often stated as being because of one or more of the following traits:

- IPsec is an open standard and therefore has no immediate cost associated with using it for further development.

- Its modular design allows the implementation of new encryption and authentication algorithms.
- IP-based addressing integrates directly with IPsec, though should alternative addressing schemes be used, IPsec becomes undesirable due to this dependency.
- Higher-level operations, such as those operating at the application layer, do not need to be modified to accommodate IPsec.
- IPsec is forwards compatible with IPv6, allowing it to operate in extremely large networks.

(Ghosh et al. 2005) propose that ad hoc networks may be secured using IPsec. Referring to the ITU-T X.509 document regarding the use of certificates for access control and authentication as a basis for a lightweight authentication and certificate revocation method, Ghosh reports that IPsec may be implemented in ad hoc networks with tolerable impact on packet latency. It is shown that IPsec can be made to work in a hop-by-hop manner, securing neighbours in a MANET, but this is argued to have a significant impact on network performance due to packets being de-encapsulated and re-encapsulated at each node. It is suggested that end-to-end services should be the focus of MANET implementations of IPsec, assuming that intermediate hops cannot be trusted. It is not mentioned which routing protocol is used to generate network topology, nor is it suggested that IPsec-like frameworks can be extended to protect routing data.

(Ali et al. 2010) suggest that modelling whole-network MANET security on existing frameworks is possible, but only if extensions to such frameworks are developed to account for the unique attributes of MANETs. MANIPsec is proposed as such a framework, providing a closed-network with authentication and access control services. As with (Papadimitratos & Haas 2006) proposal, this framework only accounts for end-to-end communication.

A key issue raised by both Ali and Ghosh is the concept of dynamic key generation. It cannot be safely assumed that a network will have pre-shared keys, or that a network wide key will remain secure should a node be physically compromised. (Puttini et al. 2003) focus on the concept of certificates as a means of providing authentication services, suggesting that despite the additional overhead that certificates cause when nodes authenticate with each other, the dynamic key generation services that they offer are

valuable. They extend this statement by suggesting that by closing the network at the node level, issues caused by the open medium of the network may be mitigated. Like Ghosh and Ali, Puttini focuses on end-to-end communication security. (Puttini et al. 2004) does expand the concept of MANET security to the network plane, specifically highlighting the need for routing protocol security and communication security to provide protection to the whole network. They do conclude, however, that although observed overheads were small, the proposed framework was only applied to small MANETs with low hop counts.

(Lacey et al. 2012) propose Reputation-Based Internet Security Protocol (RIPSEC). This framework is an alternative approach, in which the behaviour of nodes during the routing process is used to inform the IPsec component of the trust-levels assigned to a given node. This trust-level can be shared among other members of the network, to identify and counter-act malicious behaviour. However, this only provides security countermeasures to non-routing communication, routing is used to inform end-to-end communication using IPsec-like security, but is not protected.

(Kang & Balitanas 2009) highlight security vulnerabilities in the IPsec approach. They identify issues with the remote-access approach of IPsec, stating that it is possible for unauthorised devices local to an authorised device to access the IPsec connection through the authorised device. It is possible that an associated home computer, infected with a worm, may be able to propagate that worm to shared network drives through the local authorised machine's IPsec connection. The critical vulnerability highlighted here is the trust placed on end-points to have robust access control policies, despite IPsec enforcing no such policy itself. IPsec, despite providing secure communication over an untrusted medium, is vulnerable to any weak access control policy at end points and does not provide in-built solutions to this problem.

It has been observed, as recently as 2014, that some of the security algorithms used by IPsec have expired (been cracked or left the period of reasonable assumption of security) (Dadhich et al. 2014). IPsec is configurable, users can select from a variety of encryption and hashing algorithms. However, IPsec still includes Message-Digest 5 (MD5) and Data Encryption Standard (DES), in this suite of algorithms. Both of these security algorithms have been demonstrably cracked on contemporary computer hardware in a timely manner (22 hours to crack a 56-bit symmetric DES key), demonstrating their weakness. That these

continue to be included in the IPsec standard leaves end-user organisations reliant on local expertise to ensure the security of their IPsec implementation. There have also been security issues identified with IPv6 implementations of IPsec, relating to link-local Neighbour Discovery Protocol (NDP) (Supriyanto et al. 2013). This further supports assertions that IPsec is reliant on existing layer 3 network architecture, and may require substantial revision in the near future.

The combination of IPsec and secure routing protocols has been proposed in previous literature. Analysis of these approaches has frequently found issues in the sharing of secrets between nodes. (Patil & Sidnal 2013) discuss secure routing and IPsec independently and in combination as a part of their survey of secure MANET routing. IPsec is found unsuitable for the protection of MANET routing, due to its need for either pre-shared secrets to generate keys, or the presence of an online trusted third party (which is not a safe assumption for an ad hoc network). Secure routing protocols tend towards an assumption of trust, any transmission that meets the protocol standards used (and uses appropriate keys) is assumed to come from a trustworthy source. They conclude that secure routing protocols, and IPsec used to secure routing, both share issues related to the assumption of trust inside the network, and dynamic incorporation of new nodes in the field.

Combined approaches, in which IPsec secures non-routing communication and secure routing ensures that the network topology generation process is protected, cannot guarantee full communication security. If these protocols use separate tables to store credentials, attackers can compromise services individually. (Wallgren et al. 2013) identify that this does increase the workload for an attacker trying to fully compromise the network, but they do not need to fully compromise security to disrupt the network.

Compromising routing would allow for the destruction or theft of IPsec protected data regardless of if the IPsec service itself had been compromised (Von Mulert et al. 2012). Likewise, due to the open-medium problem, compromised IPsec would allow for the theft of encrypted data between end-point without any need to interact with secure routing protocols. Using the two approaches in tandem does not provide a full-security solution, it has been reported that mitigation of threats is possible, but full network security cannot be guaranteed by combined approaches (Kumar et al. 2012). It has been proposed that

hybrid, integrated approaches may provide such security, but no such proposals have been identified (to the authors knowledge) at this time.

2.6.2 MANET Focused Approaches

Frameworks have been proposed for MANET security independently of IPsec. (Wang & Teng 2013) propose an efficient authentication scheme for secure MANET communication, proposing a sliding scale of security services that may be implemented depending on the perceive threats to the network. An issue identified with such a framework is that although it is able to reduce the security overhead when the network is not perceived to be under threat, threat analysis and response represent significant control challenges in distributed systems. (Jang & Agha 2006) propose a framework describing the means by which communication overhead may be reduced, providing resources that may be used for threat-analysis by reducing the cost of other network services.

It is possible that attacks could be made against the threat analysis portion of the network. Increased latency and reduced bandwidth result from the most secure states of the proposed framework, allowing attackers to potentially manipulate the network security service by increasing or decreasing the perceived security threat to a node in a calculated manner. To mitigate these issues a cluster-based topology is proposed, similar to that of (Rachedi & Benslimane 2006). This is essentially a requirement for the reliable operation of the proposed framework in large MANETs.

(Lu et al. 2013) report on their secure distributed authentication scheme based on Chinese Remainder Theorem – Verifiable Secret Sharing (CRT-VSS) and trusted computing, combining the concepts of closed-network and trust-based routing. The need for protection of the key generation and exchange mechanisms in MANET is identified, with misbehaviour during the authentication of nodes being identified as a potential point of ingress for malicious nodes possessing partial network information (locations of nodes and known plain-text). It is argued that by establishing trust between nodes, in addition to providing authentication services, the authenticity of nodes may be re-evaluated as

required by the network to protect against identity-based attacks which aim to abuse the established trust between authenticated node-pairs.

This is similar to the peer-evaluation mechanism of TRUNCMAN with regards to how trust is computed by neighbouring nodes, not the network as a whole. Communication complexity is reduced, decreasing the control overhead of routinely checking for misbehaviour when nodes attempt to authenticate with each other. As identified in previous work by (Glynos et al. 2005) regarding the prevention of impersonation attacks, the proposed framework only protects the authentication of nodes. It does not extend security services to routing or communication security directly, instead assuming that a closed-network will provide a degree of protection to those services.

2.6.3 Section Summary

IPsec provides an open, robust and standardised framework for security, but is restricted to infrastructural networks, for the most part. Proposals for the integration of IPsec into MANETs have been made; finding that although the impact on communication overhead is tolerable, the networks tested have been relatively small. Attempts to implement IPsec directly in larger networks have found that it does not scale well with large numbers of nodes and high hop counts.

Bespoke security solutions have been proposed, but tend to focus on specific MANET attributes, such as protection of routing information. These approaches tend towards end-to-end security rather than accounting for the lack of security between source and destination nodes in multi-hop scenarios.

This research has identified current approaches to developing and deploying MANET security frameworks. IPsec remains popular as a foundation for continuing work, but many methods still focus on end-to-end security implementations that ignore the unique attributes of MANETs. Current solutions have not proven themselves scalable, being tested on MANETs consisting of no more than twenty nodes.

2.7 Research Gap Analysis

The literature review has outlined a distinct area of research that may be sub-divided into two main areas: DTA and MANETs. Publications regarding DTA generally reference a communication method, with the majority specifying that due to the mobile nature of nodes in the mission area, wireless communication over a MANET is desirable. Research into MANETs has suggested their utility in autonomous systems, due to their ability to allow communication in highly changeable environments, such as those likely to be encountered by UAV systems. In both cases, the suitability of DTA over MANETs has not been investigated. The communication requirements of autonomous decision making over MANETs is an open problem and a suitable area for additional investigation.

The security of MANETs relates directly to the subject of autonomy. If a system is to operate without human intervention, it must have some means of security to protect against attacks which seek to compromise it. MANET routing security has been analysed and many solutions proposed in contemporary literature, with behavioural and cryptographic defences specific to the needs of MANET architecture having been identified. There is, however, a trend to focus on network and control communication as entirely separate. This represents another avenue of investigation, as data security in MANETs is an area of great interest, but MANET security tends to be viewed as routing specific.

The following gaps have been identified in the research reviewed in this chapter:

- DTA's demands on MANET resources may be high, due to the communication complexity of fully decentralised task allocation algorithms. MANET constraints and the attributes of wireless communication over radio will have an effect on the timeliness and optimality of DTA assignments. There is a research gap that has been identified in Sub-sections 2.2 and 2.4 regarding the impact of communication constraints on DTA processes.
- Security has been identified throughout sub-sections 2.3 and 2.5 as a critical issue. MANET routing security has been identified as an issue in the logistics of information exchange: insufficient security in this area can result in the subversion of the routing protocol by malicious nodes, preventing the reliable delivery of

messages in the network. A gap has been identified relating to the open-medium problem of wireless communication; investigating and proposing a means of preventing trivial access to network-related information.

- Sub-section 2.5 reviews literature related to the issue of dynamic key generation. Tolerating the arrival and departure of nodes in a MANET has been identified as vital for missions with durations exceeding the operational lifetime of the selected platform. Investigating a means of providing security, while allowing nodes to join the network after initialisation, is an open problem and research gap when presented in the context of autonomous MANETs.

The above points identify the research gaps which the thesis will continue to analyse and identify solutions for.

2.8 Chapter Summary

This literature review has collected a wide-range of publications from a broad base of research. Key attributes, requirements and issues related to MANETs have been identified, with state of the art proposals and solutions identified for autonomous control (DTA), routing, communication and security. Existing research that can provide a foundation for understanding the interactions between MANETs, autonomous control and security has been identified. Furthermore, existing security frameworks have been identified, which can assist in identifying the requirements of security and provide a basis of comparative analysis for the research outcomes of the thesis.

3 PROBLEM ANALYSIS

3.1 Chapter Introduction

The literature review has provided an overview of the research domain and a foundation of existing literature from which original contributions have been identified. This section will analyse the research questions posed in Sub-section 1.2; identifying key areas of investigation, approaches and tools of potential benefit to the research, and defining the scope of the research to be undertaken.

By the end of this chapter, the problem domain will be defined by a set of hypotheses. These hypotheses will be formulated from an analysis of the existing literature, the setting of an appropriate research methodology and areas for investigation. This will provide a framework for the research to be undertaken, defining clear goals and the general requirements that must be met to achieve those research goals.

3.1.1 Chapter Layout

This chapter is broken down into the following sections:

- Section 3.2 analyses the issue of autonomous control, specifically the communication requirements of DTA algorithms.
- Section 3.3 looks at the problem of providing security in highly verbose networks, such as those performing DTA to support autonomous activity in a MANET.
- Section 3.4 discusses the fundamental assumptions made when planning the research.
- Section 3.5 consolidates the findings, defining the research scope and proposing hypotheses that will guide the research reported in this thesis.
- Section 3.6 summarises the chapter.

3.2 Distributed Task Allocation (DTA)

As discussed in sub-section 2.4, DTA algorithms provide a means by which a network of nodes may cooperate to solve complex problems. It has been identified, however, that these algorithms have several limitations that are especially apparent in the context of MANETs, namely the complex communication required to arrive at a network-wide solution, and the time taken to arrive at that solution.

DTA, as discussed in the literature review, may be centralised or decentralised in nature. Due to the dynamic mobility of MANETs, however, centralised approaches can restrict the movement of nodes within the mission area, as hop counts are constrained to preserve quality of service and nodes are required to be within a given range of the central authority for task allocation. Centralised approaches also have a single point of failure, making them undesirable in MANETs with energy limitations, or a possibility of node loss.

Consensus-based approaches in particular present a variety of algorithms that allow nodes to move freely within the mission area, while allowing nodes to participate in task allocation with fewer constraints on mobility when compared with centralised approaches. Such approaches also allow the computation of solutions with limited computational resources by only requiring that nodes compute locally optimal solutions until a compromise is reached between all such local optimised solutions and a global solution is formed.

Decentralised approaches to DTA avoid the single-point of failure issue by only losing the resources of a given node, when a node is removed from the network, instead of losing the ability to control the DTA process entirely. This makes such algorithms attractive for use in scenarios in which node loss is a possibility, such as hazardous materials inspection or search and rescue.

Consensus-based bundle algorithm (CBBA) has been identified as a means by which a collection of nodes may compute globally optimal solutions in an optimal manner. Although it allows solutions to be computed taking into account the whole network, it suffers from complex communication that is ill-suited to the network architecture in which it is more frequently employed. CBBA requires that nodes communicate in a highly sequential manner, which is at odds with the parallelism offered by MANETs.

Sequential communication in a MANET can be described in terms of a series of unicast transmissions. Though these transmissions are addressed to specific nodes, they are still inherently broadcast due to the nature of wireless communication. Assuming that the medium for communication is wireless, all (even addressed) communication will propagate outwards in all directions from the point of transmission. Many nodes may be within range of such a transmission, but if the communication is addressed to a specific node, the other nodes in range will ignore it. This may represent a significant amount of wasted communication over the course of a task allocation process.

3.2.1 Defining Communication Cost

Cost must be quantified to accurately represent its impact on a system. In the case of communication cost, two key factors play a role; the number of communication events and time taken to communicate. Communication events represent individual transmissions regardless of the number of packets involved, representing the communication required to share a bundle in the case of CBBA. The time taken to communicate is a direct measurement of the amount of time that elapses between the beginning of DTA related communication and the end of that round of task allocation. The sum of all such measurements leading up to achieving consensus is the total communication cost of a CBBA process.

When referring to waste, it must be noted that communication perceived to be redundant is classified as waste. Costs are not inherently wasteful, but the repetition of communication that may have been transmitted more efficiently is described as a waste cost in the context of this research.

3.2.2 Communication Complexity

The sequential nature of communication under CBBA is a major issue when considering its use in wireless networks, but the complexity of that communication is another key factor to consider. Equation 3-1 (Johnson et al. 2010) demonstrates the growth in

communication complexity as a network grows, when performing CBBA. *x*, being the number of communication events required to reach consensus, is dependent on the number of nodes, *n*, in the network and the function of the number of CBBA rounds required to reach consensus, *r*.

$$\boldsymbol{x = f(r).(n.(n-1))} \qquad \textbf{(Equation 3-1)}$$

As a result, communication complexity increases rapidly as the number of nodes involved increases. This limits the scalability of CBBA, making it unsuited to large networks.

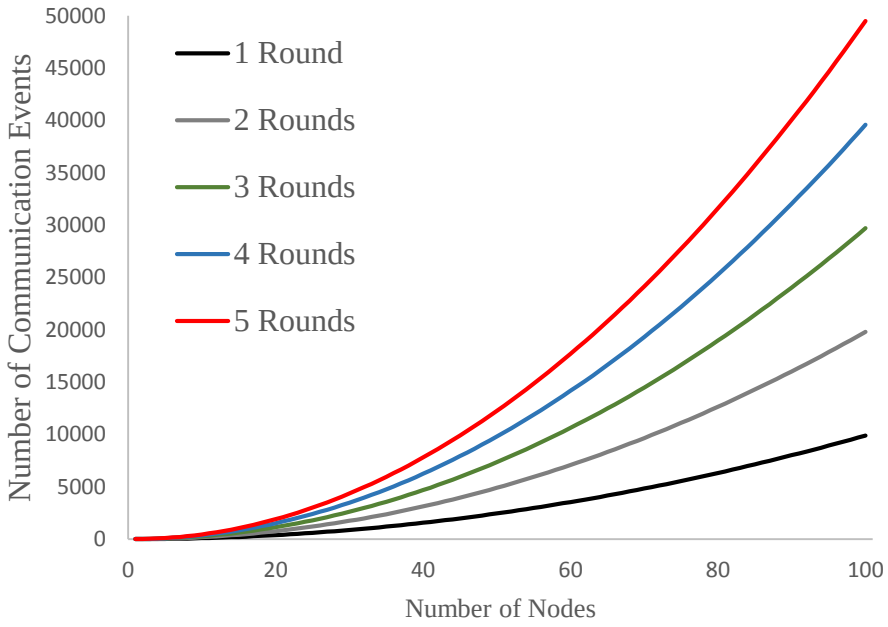

Figure 3-1 Graph showing the effects of network size and number of rounds on the required number of communication events

Figure 3-1 shows the effect of the number of nodes on the number of communication events required to complete a variable number of CBBA rounds. It is assumed that bundles do not exceed the Maximum Transmission Unit (MTU) of the network interface. It becomes apparent that when solving complex problems, CBBA will place a significant demand on network resources to achieve consensus. This is primarily due to the sequential nature of communication between nodes under CBBA.

CBBA requires that each bundle transmission is synchronised across the network. Early implementations of the algorithm used sequential communication to ensure that nodes would have all received bundles from the current source node and performed CBBA on them (Brunet et al. 2008). This was achieved by ensuring that the next node to communicate its bundle was the last in the sequence of unicast transmissions by the source node. Current implementations of CBBA have not reported on the communication used to facilitate MANET functionality, though multicast and broadcast communication have been identified as possible and potentially beneficial with the addition of control traffic to allow synchronisation of nodes receiving broadcast bundle updates (Colistra et al. 2014).

The number of tasks increases the computational complexity of problem solving in a linear fashion, but unless the size of packets required to communicate bundles exceeds the network interface MTU, this will have no effect on the communication complexity of the assignment problem.

Due to the sequential nature of the communication, strain may also be placed on reactive routing protocols if the time taken to communicate with all other nodes in the network exceeds the lifetime of a route. Such an occurrence would force the network to plan a route for every unique communication, unless routes were consistently utilised within their lifespan and nodes did not move in such a way that such routes become invalid (in the case of reactive protocols such as AODV). Similarly, a network undergoing rapid topological change due to high mobility would suffer under reactive routing protocols, as well as proactive protocols with an insufficient polling rate.

This combination of factors suggests that timely communication is required, and the means by which communication may be made timely is the reduction of complexity. By harnessing the inherent attributes of MANETs, such as broadcast communication and node-based routing, it is possible to reduce communication complexity. Broadcast communication allows multiple nodes to be addressed with a single transmission, while the node-based routing of a MANET can allow reconfiguration of the network into clusters as needed. By reducing communication redundancy and sub-dividing the task allocation problem, the complexity of both the CBBA process and the communication required to reach consensus may be reduced.

3.2.3 Section Summary

Key issues have been identified in terms of CBBA's communication complexity. It is possible that communication related to CBBA task allocation may take a significant length of time in large networks due to the sequential nature of its communication. If no other activities are being undertaken during this time, it is possible that some nodes, such as UAVs, will be expending resources to remain mobile without actively pursuing mission objectives while waiting for the results of a CBBA process.

The result of this analysis is the observation that the problems associated with CBBA may be categorised as follows:

- Task allocation: The task allocation problem may be divided and allocated to clusters of nodes. This is possible due to the high-degree of control MANETs have over the description of their topology, allowing nodes to be given varying privileges and responsibilities when performing DTA (such as becoming a cluster head). This may be done with little impact on network services and topology; initial control communication required to designate node roles being the only additional required communication.
- Efficient use of the communication medium: Harnessing the inherently broadcast nature of wireless communication will cut down on the waste inherent to sequential, largely redundant transmission of bundles between nodes. By allowing multiple nodes to be addressed in a single transmission, the number of repeated messages will be reduced, in turn reducing the complexity of communication and the time spent communicating.

To address these two issues, a performance analysis of CBBA would need to be undertaken. This would inform the proposal and development of a MANET optimised version of CBBA including the previously discussed modifications to cut down on communication complexity and the time taken to reach consensus in MANETs of varying size.

3.3 Network Security

Security should be considered in terms of authenticity, integrity and confidentiality. The inherently trusting nature of MANET nodes leads to issues with authenticity, as without a means of identifying legitimate and malicious nodes the network is vulnerable to attacks that abuse the implicit trust between nodes. The integrity of packets sent through the network is an important consideration, as packets lacking integrity assurance measures may be modified in-transit to their intended destination. Confidentiality mitigates the likelihood of an outside observer being able to derive valuable information from captured packets, at least in a timely manner.

Existing proposals for MANET security focus on security as a routing problem or control problem, providing protection to routes or data. Such approaches view the integrity of routes and the confidentiality and integrity of control data independently, with proposals such as SAODV and SOLSR addressing MANET routing security and MANIPSEC focusing on end-to-end data security. An extensive literature review did not identify approaches that address whole network security. To better understand the need for security that can be applied to the data communicated through a MANET, the types of attacks commonly used against MANETs must be analysed. The open-medium problem must then be analysed to identify the steps that must be taken to address the issue of inherently broadcast communication.

3.3.1 MANET Vulnerabilities

MANETs are vulnerable to a variety of attacks, primarily due to the following reasons:

- Nodes may become isolated, allowing malicious nodes to exploit reputation mechanisms or the lack thereof to communicate with such nodes.
- Wireless communication occurs on an open-medium, allowing trivial interception of messages, which if not encrypted, are easily read. If messages are not authenticated and integrity checked, modification of packet contents is possible,

leading to potentially unreliable or harmful communication which appear legitimate.

- MANET nodes implicitly trust each other, leading to potential abuse if malicious nodes appear to be legitimate (or if the MANET nodes do not categorise participating nodes).

Figure 3-2 shows an instance of communication between two end-points via an intermediate node in a MANET. A route has been formed between the two nodes, requiring an intermediary to relay messages between source and destination.

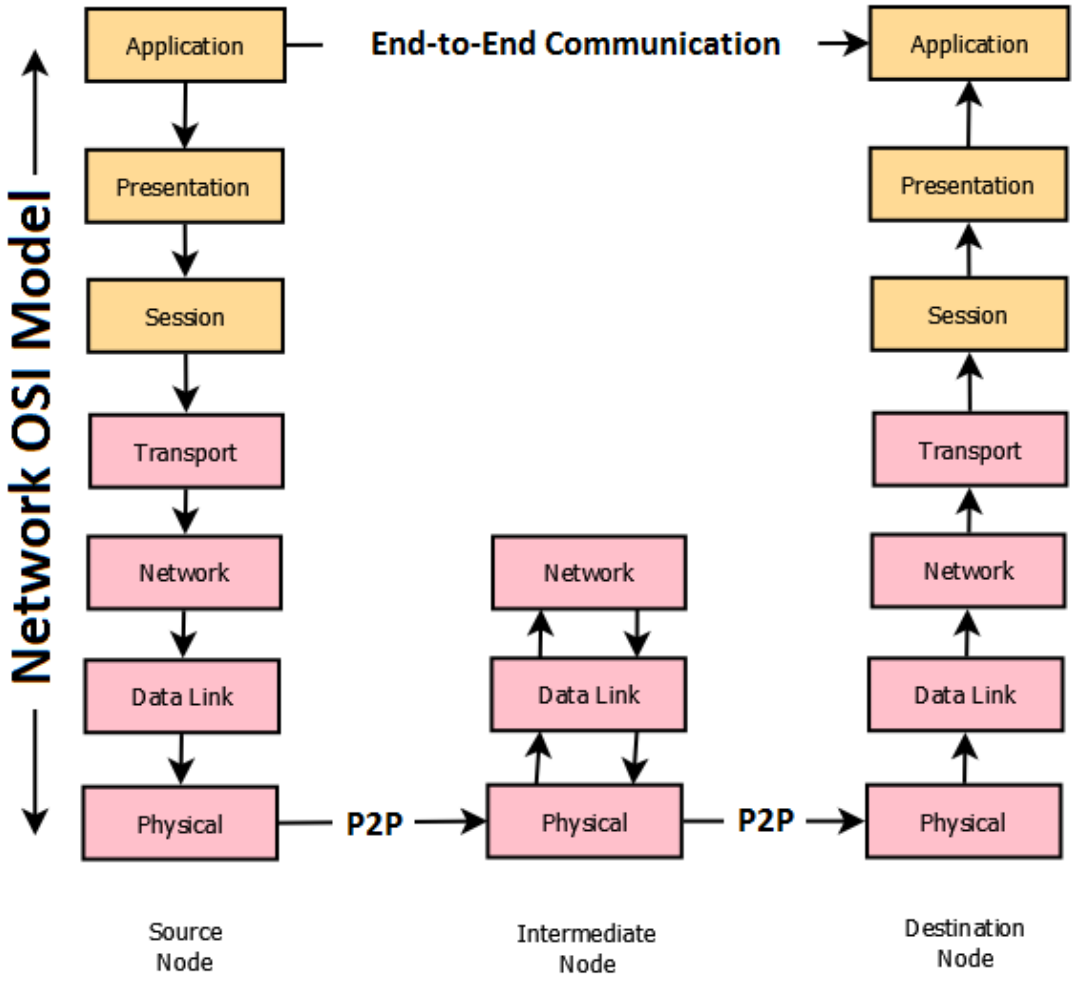

Figure 3-2 Data flow between two MANET end-points via an intermediate MANET node (OSI model)

End-to-end communication, in this thesis, is the transmission of information between two nodes regardless of route taken or number of hops. It can be seen that end-to-end communication is focused on outcomes; and application or service on the source node communicates information to the destination node to inform or drive an application there. However, to facilitate this process, data must be passed down the network stack, through the network layer (which provides the route to be taken and next node to send the packet to) and data link layer (which provides an appropriate addressing protocol). At the physical layer, the packet is transmitted, starting the first instance of point-to-point communication. It is at this point that the network becomes vulnerable.

Again, the packet becomes vulnerable, as a node between the intermediate node and destination may listen in on the packet. Real time modification of the packet is non-trivial, but observation of packets can still provide an attacker with information about the network and services running on it. As the destination node is within transmission range of the intermediate, it is extremely unlikely that any manipulation of the packet can occur prior to receiving the legitimate transmission, though replay attacks may still be possible.

This demonstrates that the intent, meaning and effects of end-to-end communication can be manipulated or destroyed by an outside attacker by attacks that target point-to-point transmission of information throughout the network. As end-to-end communication is vital for distributed control and the provision of services across the network as a whole (instead of only among local neighbours), it is vital that the types of attacks that may be launched against the network be identified, so that appropriate countermeasures may be proposed.

3.3.2 External Attacks

(Chandra 2005) categorises external attacks against MANETs as follows:

- Passive attacks, such as packet sniffing, data collection and observation of nodes.

- Active attacks, such as manipulation of data and destruction of packets being sent out of the MANET in question.

Passive attacks are extremely difficult to detect, as they do not require malicious nodes to make themselves known or participate in the network in any way. Precautionary measures must be taken to hide vital information from passive observers, as it is impossible to prevent the potential capture of packets due to the open-medium problem.

Active attacks will make the malicious node known to the network, but only if nodes can be identified and authenticated. An active attacker may capture and replay packets, or modify data that is being sent back towards a controlling authority, such as a base station. It is possible that a MANET may have to periodically transmit data back to a base. Search and rescue operations are an example of such a scenario, where a MANET in the field is used to gather data autonomously and provide information to human observers at a remote base. The modification of data between a base and MANET is unacceptable, as mission critical information may be modified or destroyed. Such attacks are collectively referred to as man in the middle attacks. These attacks are of concern to MANETs, due to the fact that each node must operate as a router as well as an end-point. As a result, malicious nodes may attempt to insert themselves into the network by capturing and retransmitting communication between distant nodes, via unsecure routes.

3.3.2.1 Man in the Middle

Man in the middle attacks are a class of attack that does not require direct participation in unsecured communication. They may be used to gather intelligence, abusing the inherent trust between MANET nodes to insert themselves into the area of operations, recording and relaying packets between legitimate nodes. These attacks can operate in two modes; active eavesdropping, in which messages are intercepted and relayed (without modification) to their intended destination, and malicious modification of packets before forwarding them to their intended destination.

This attack exploits the inherent trust between nodes in a MANET, trust based on assumptions of cooperative behaviour and legitimacy in networks that do not have security mechanisms in place to provide node authentication. In MANETs that do not

provide integrity and confidentiality services to packets, capture and modification of communication is trivial. A common goal for such attacks is the bypassing of network defences via weak points in network security to lay the foundations for internal attacks with a wider scope.

3.3.3 Internal Attacks

Internal attacks can occur only when network perimeter defences, such as access control and authentication, have been compromised or simply not been implemented at all. In the case of MANETs, access control and authentication services are not provided as standard and many security protocols only seek to secure one aspect of the network, be it routing, control or other data.

These attacks can have a considerable impact on the network, as they have access to key network services due to their direct participation in the network. Two key attacks relevant to MANETs are analysed in this thesis; masquerade and packet spoofing.

3.3.3.1 Masquerading

Masquerading can be considered to be a two-stage attack; a malicious node adopts the identity of one or more legitimate nodes and then abuses its apparent legitimacy to further whatever goals the attacker may have for disruption or destruction of the network.

Due to their ability to act as if they are legitimate nodes, masquerading nodes may also abuse certain trust-based security systems, such as those implemented in TRUNCMAN (Thanigaivel et al. 2012) routing protocols. By broadcasting false reports of neighbouring node misbehaviour, the attacker can cause legitimate nodes to be flagged as potentially hostile, reducing the effectiveness of the network by triggering resource intensive defence mechanisms or causing nodes to be dropped from the network. This is commonly used as a means of allowing more malicious nodes to enter the network, as the malicious node may vouch for these additional attackers. The more malicious nodes that enter the

network, the greater the control of the attacking party, especially in autonomous systems that rely on cooperation.

The Sybil attack is an example of a masquerade attack based on identity spoofing, first identified by (Douceur 2002). The objective of this attack is to obtain as many node identities as possible to abuse data redundancy features and stored information, such as routing tables, to inform an attacker of network characteristics and allow direct intervention in the network under the guise of stolen legitimate identities. It specifically targets peer-to-peer networks and subverts reputation systems to undermine security and ensure its inclusion in network activities.

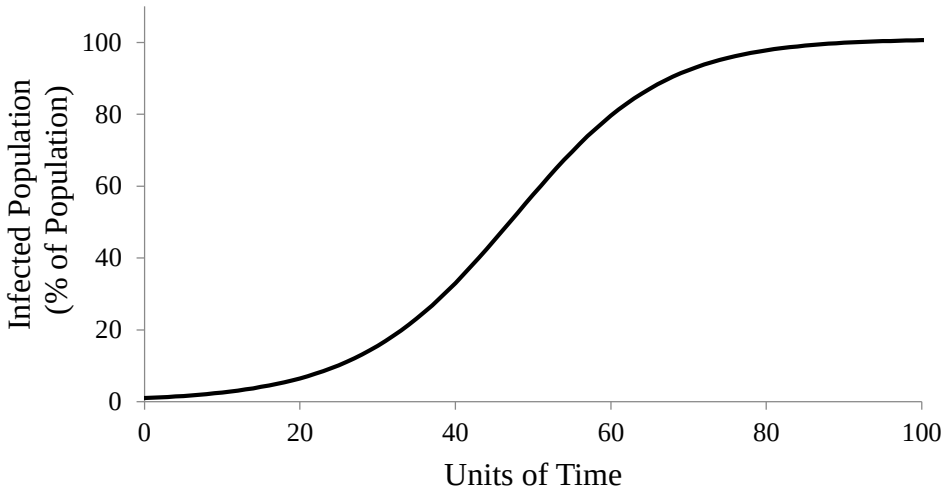

Figure 3-3 Graph showing the rate at which nodes become untrusted due to malicious activity (Bulygin 2007)

Figure 3-3 provides a visualisation of the impact of identity spoofing on a MANET, using the Susceptible Infected (SI) model as applied to random scanning attacks against networks as described by (Bulygin 2007). Assuming no security countermeasures and an open-network, a malicious node masquerading as one or more legitimate nodes may quickly cause those nodes to become untrustworthy. If the network has a means of blacklisting nodes, this will merely remove them from the network, depriving the MANET of resources as nodes are ignored. In networks without trust systems that track the reputation of a node for misbehaviour, the line in the graph can instead show the

number of nodes which may no longer be trusted to act in a legitimate manner, though an autonomous MANET will be unable to identify misbehaviour in such cases.

3.3.3.2 Packet Spoofing

Packet spoofing involves the creation of packets designed to be accepted as legitimate by the network. This may be due to the perceived legitimacy of the transmitting node, but may also be injected into the network on an individual basis, not requiring a persistent malicious presence in the network.

In networks without integrity checking and authentication services, packet spoofing represents a significant threat to all network services, as false data may be introduced to the network and treated as if it were from a legitimate source. This can lead to abuse of routing protocols or provide a means of entry for nodes intending to perform identity-based attacks such as the Sybil attack.

Simpler attacks such as Denial of Service (DoS) attacks are also possible, as packets that appear legitimate will not be discarded as quickly as those perceived as malicious. The longer a packet persists in the network, the more network and processing resources are consumed. It is possible to severely reduce the performance of a network or shut it down completely by saturating the communication medium with packets, but if those packets appear to be legitimate, far less effort is required on the part of the attacker.

Equation 3-2 (Baskett et al. 1975) mathematically models an M/M/1 queuing system, which can accurately represent the queuing systems at work in a MANET, in this case a line topology of nodes relaying packets to each other. *E(q)* is the total expected response time of a node, γ is the arrival rate of packets at a node, and ρ is the result of arrival rate divided by service rate.

$$\boldsymbol{E(q) = \frac{\rho}{\gamma(1-\rho)}} \qquad \textbf{(Equation 3-2)}$$

Figure 3-4 provides an example of the effects of increased arrival rate on the ability of a node to process packets. Malicious packets can increase the rate of arrival, forcing a node to process them as if they were legitimate, increasing the response time of the node.

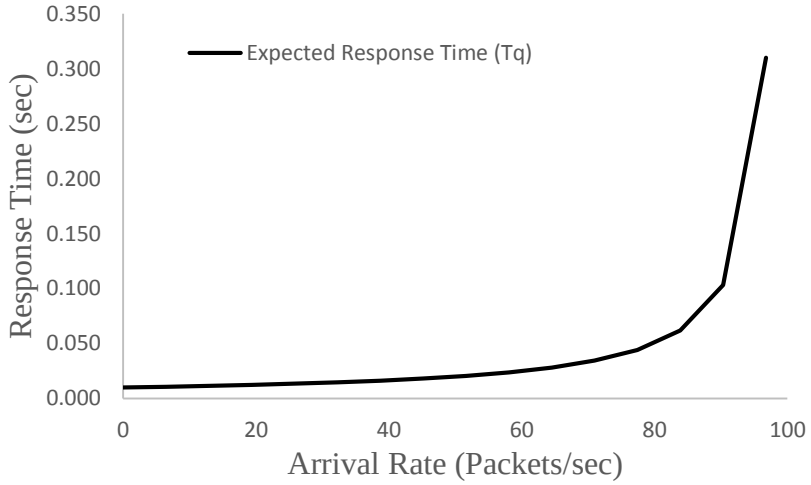

Figure 3-4 Graph showing the effects of arrival rate on the expected response time of a node

Increases in response time reflect a degradation of the networks ability to handle traffic. As the arrival rate gets closer to the capacity of a node, the likelihood of needing to use a sub-optimal route increases. This effectively increases the number of retransmissions required to communicate a single packet, adding more load to the network globally as a result of all local additional transmissions required to maintain service provision.

Furthermore, all traffic that does not directly contribute to network or control services wastes the finite resources available to nodes. If security measures that allow the expedient removal of malicious or wasteful packets from the network are not enforced, an attacker may impact quality or service or possibly the ability for the network to provide vital services at all. If an attacker is able to generate more traffic than strategically vital nodes (those with many links which form a nexus for communication across the network) are able to process, a race condition may be generated, in which the affected nodes continually process wasteful packets or deliver unacceptable wait times for the processing of legitimate traffic.

3.3.4 The Open-Medium Problem

The previously discussed attacks are a particular problem for MANETs due to a vulnerability referred to as the open-medium problem. MANETs commonly use wireless communication to provide a medium for the network. Wireless communication is broadcast, a term which has multiple meanings depending on the context in which it is used. In the case of wireless communication, control of directionality and the range to which communication may propagate is difficult to control. As a result, control of the medium itself is non-trivial and it may be resource and hardware intensive to attempt to control.

A packet may be transmitted in unicast, multicast or broadcast mode, but the transmission itself propagates outwards from a transmitting node in all directions unless specific hardware is used. Even if directional antennae are used, these must be angled at the correct nodes to ensure they receive transmissions, leading to further control complications as nodes must be aware of each other's positions and directional communication may increase the need for sequential communication in services that leverage the broadcast nature of wireless communication, such as MANET routing protocols.

This is referred to as the open-medium problem, the propagation of the transmission itself cannot be controlled without the use of specialist hardware or by changing the transmission power of the node. When considering lightweight UAVs and similar platforms commonly used as MANET nodes, specialised hardware may not be feasible due to payload weight or the mechanical and electronic complexity of such devices.

Figure 3-5 compares the effectiveness of three different approaches to the issue of identity attacks in an autonomous MANET. With no counter measures, an identity attack randomly assuming identities until it obtains information allowing it to fully assume the identity of a legitimate node will propagate through the network as shown. This reflects it being able to reach more nodes by assuming the identities of their neighbours, or gaining greater control of network and control services.

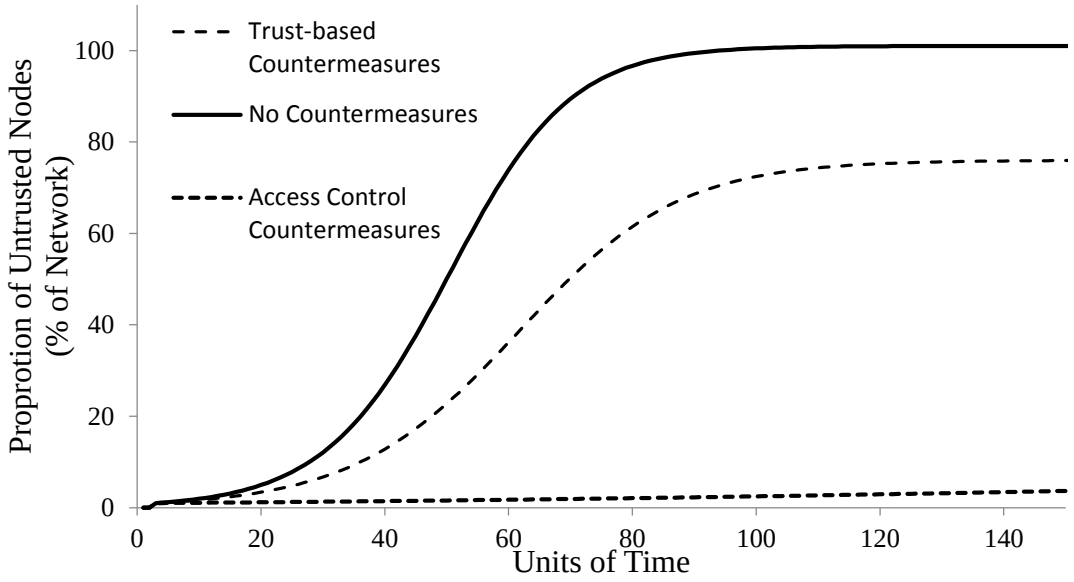

Figure 3-5 Graph comparing the effects of countermeasures on the rate at which nodes become compromised or infected by an attacker (Zou et al. 2003)

Trust-based countermeasures that allow the blacklisting of misbehaving nodes may slow the rate at which identities are subverted, possibly even preventing it if nodes are identified fast enough. However, abuse of trust mechanisms is possible, and the behaviour observed by a passive attacker while a MANET attempts to fend off a masquerade attack can inform the active attacker of alternative strategies to abuse or circumvent security measures.

3.3.5 Section Summary

Analysis of MANET vulnerabilities has identified the following issues:

- Wireless communication, using non-specialist radio as a medium, is open and may be freely observed by nodes outside of the network.
- The ability to gather data passively, avoiding detection or direct intervention in the MANET can allow attackers to profile the vulnerabilities of any security implementation.

- The inherent trust and cooperation requirement of MANET routing protocols leaves them vulnerable to attacks that abuse the implicit trust that nodes place on planned routes.
- Secure routing does not extend security to data transmitted over those routes.

An autonomous MANET is therefore vulnerable to attacks against its routing mechanisms and task allocation algorithms. Disruption of either service may compromise the network's ability to perform the assigned mission, incurring additional costs in time and resources. Should a network be terminally compromised, the mission will automatically fail.

The need for security that is applied to each node in the network, instead of relying on infrastructure like firewalls and secure gateways, has been identified, as the topology of a MANET is subject to rapid change. Each individual node is a router as well as an end-point in a MANET and so must provide security services similar to those expected of infrastructural networks. This applies to all nodes, due to the open-medium used for communication preventing the effective use of heterogeneous security that varies between nodes.

3.4 Assumptions

3.4.1 All nodes have identical specifications

All nodes are assumed to be light-weight, aerial platforms with identical hardware. Communication, computation and mobility are uniform across the swarm and no additional payload hardware is considered in the scope of this research. The research focuses on the cost of communication between nodes, not on the intricacies of heterogeneous-network task allocation, making the consideration of such networks outside the scope of the research.

3.4.2 DTA is performed while nodes are immobile, and derives a single solution

It is assumed that nodes are static when cooperating to find a solution to a DTA problem. This assumption extends to key negotiation and authentication of nodes. The rationale for this assumption is that:

- During DTA, as position is vital to the bids placed, movement will naturally degrade the accuracy of a round of allocation (Ren & Beard 2005). This is because nodes may have moved since the previous iteration of that round. This could lead to scope-creep and acceleration away from convergence. As a result, DTA is simulated with static nodes.
- Key exchange or generation, may in some cases be location-aware (Yang et al. 2012), which would lead to the same rationale for assuming static nodes, as the point above.
- Assuming location independent key exchange/generation, the process can be assumed to occur so quickly (in the order of milliseconds including radio propagation and processing at both ends), that displacement of nodes relative to one-another is minimal. As such, it is safe to assume that nodes are immobile.

3.4.3 Communication occurs with no loss of packets or disruption

Communication between nodes is assumed to be performed on a perfect channel. There is no wasted bandwidth or loss of data on the air. As the research will focus on a comparative analysis of existing and novel approaches to DTA and MANET security, comparing communication cost (bandwidth use and number of transmissions), the use of a loss probability variable is considered unnecessary.

Such a variable would affect existing and novel approaches identically, as communication range and route lengths would remain constant in an iteration of simulation (Ren & Beard 2003). This would result in equivalent loss of data under each approach, resulting in the same percentage difference in cost between existing and novel approaches. This is

because none of the security and DTA approaches documented in this thesis modify underlying routing protocols, data-link layer or physical layer protocols.

3.4.4 Nodes are equipped with non-directional wireless transmitters

It is assumed that the radio transmitters used by nodes in this research are omnidirectional, with symmetrical propagation in all dimensions. A simple radio propagation model is used in simulation, drawing an absolute maximum transmission distance from a node. Nodes within this sphere are in range, those that are not are out of range. The rationale for the assumption that communication occurs on a perfect channel (Sub-section 3.4.3), holds for this assumption.

3.4.5 Constants, such as security credentials and task lists, are communicated prior to deployment

Where an authority is required to provide security constants, be they data for key generation or cryptographic keys, it is assumed that such data is provided prior to deployment. As a result, the initialisation of nodes with identities and task lists is not simulated, as this is assumed to occur in a secure, likely wireline environment.

It has been identified that a critical vulnerability in many security approaches is the initialisation phase (Kumar et al. 2012), and as a result it is assumed that direct human oversight of the initialisation phase is required. (Garg & Mahapatra 2009) suggest that an offline approach to initialisation would offer the greatest level of protection (assuming ownership of all hardware present during node set up), it is assumed that initial set up of nodes occurs outside of the mission area.

This does not mean that nodes begin the mission with full knowledge of the other members of the network. Multiple waves of nodes, or individually initialised nodes from different trusted authorities may be present when the network forms. The constants assumed to be provided prior to mission deployment are:

- A node identity, including network address and capabilities.
- In the case of secure routing protocols, appropriate symmetric keys.
- In the case of existing and novel security frameworks, data required for key generation (or keys if pre-generated).
- Certificates, tokens or other security data.
- For DTA, a task list containing all tasks for a mission (one list for one solution).

3.5 Research Scope

Having analysed the requirements of autonomous MANET control and security, and established the assumptions inherent in the thesis, the scope of the research can be defined.

3.5.1 Problem Domain Boundary

The scope of this research lies in two critical areas; the optimisation of DTA to meet the network constraints of an autonomous MANET, and the investigation and proposal of a security framework that will allow the network to operate independently, reliably and securely with limited network resources.

3.5.1.1 Optimising Distributed Task Allocation

Optimising DTA in MANETs is considered to be critical to the research for two reasons; expedient communication of a solution can save resources across a variety of MANET node systems, and the network constraints of MANETs make the side-by-side implementation of DTA and security difficult.

Therefore, the scope of the research with regards to DTA optimisation for MANETs can be described by the following objectives:

- An investigation of the network resource requirements of CBBA will be conducted.
- A method utilising the unique properties of MANET architecture and self-organisation will be proposed and analysed.
- An extension of this method focusing on the properties of wireless communication will be proposed and analysed.

3.5.1.2 Investigating MANET Privacy and Communication Security

The required network security services to provide a closed MANET have been identified and analysed. The research will focus on the proposal and analysis of an optimal security framework for autonomous MANETs, protecting all data communicated over the network. The primary areas of investigation for this aspect of the research have been identified:

- Methods for closing the network on a node-by-node basis will be proposed and analysed.
    - A means of protecting the network against outside observation must be identified, to prevent the collection of data that may prove useful to an attacker. This may be achieved by implementing confidentiality measures.

- A security framework will be proposed and analysed, with the aim of demonstrating that a closed MANET approach to security will provide adequate protection to independent, autonomous MANETs.
    - Integrity checking in a point-to-point manner is required to prevent the propagation of malicious packets through the network. End-to-end integrity checking is not sufficient, as this may still allow malicious nodes to propagate along the intermediate nodes in a route, wasting network resources and degrading quality of service between source and destination.
    - The robust and reliable implementation of confidentiality and integrity requires that the network be treated as a closed network. Methods of achieving this in an inherently open-medium must be identified and

analysed. Access control and authentication services will form the foundation of a closed MANET, and therefore appropriate implementations must be identified.

- The impact of security on network resources will be investigated, to ensure that the proposed framework considers the limitations of MANET resources and reliability.

3.5.2 Hypotheses

The focal points of the research can be expressed as a series of hypotheses. These hypotheses relate to the four objectives identified in Chapter 1 of this thesis, and are contextualised by the analyses undertaken in this chapter.

- CBBA's cost in number of transmissions and bytes communicated can be lowered by dividing the MANET into clusters.
- Multicast communication, with additional communication to facilitate CBBA's synchronisation over the network, will reduce the number of transmissions and bytes required by DTA.
- Enforcing rigorous access control policies on all nodes in a MANET will mitigate the open-medium problem.
- Allowing authenticated nodes to service authentication requests on behalf of their peers (if they share a route), will reduce the effective length of the route between the requesting node and the target node.

3.5.3 Section Summary

The scope of the research has been established, based on the analyses conducted throughout this chapter. Hypotheses have been proposed, which direct the investigations that must be undertaken to address the areas of original contribution identified in Chapter 1 of this thesis, and the four component elements of the research related to these contributions.

3.6 Chapter Summary

This chapter has analysed the communication complexity of DTA and the security requirements and unique architectural attributes of MANETs. These analyses have allowed the identification of areas of interest that will form the foundation of the research in the following chapters.

It has been identified that consensus-based DTA is globally-optimal, but highly sequential when considering its means of communication. By reducing the communication complexity of DTA, the demands of consensus-based algorithms on network resources may be reduced. The resources freed may be reassigned to other network services, such as security.

MANETs using wireless communication suffer from the identified open-medium problem. By addressing security at the node level, a MANET may become a closed-network using an open-medium. By providing confidentiality and integrity services to all data, passive attacks against the network may be mitigated. This in turn reduces the likelihood of an active attack against the network being successful.

This chapter has identified that a whole network security solution, including optimised control communication and MANET-specific security services, is required to ensure that an autonomous MANET is capable of operating independently and securely.

4 OPTIMAL DISTRIBUTED TASK ALLOCATION FOR MANETs

4.1 Chapter Introduction

Having identified the communication requirements and limitations of CBBA in Chapter 3, potential solutions may be discussed. Autonomy requires that DTA be performed, and CBBA has been identified as an effective solution to that particular problem. The need for CBBA to allocate tasks places additional demand on network resources, which needs to be quantified to account for or mitigate this additional communication requirement.

This chapter defines the methodology used to identify potential solutions, and the proposals resulting from analysis of the following:

- The communication cost of DTA, defined as transmissions and bytes transmitted.
- The need for synchronisation of the global network state under CBBA, after each bundle exchange.

By addressing these issues, methods of performing CBBA in compliance with the limitations of lightweight MANET platforms can be defined. In turn, the reduction of communication cost will allow for the addition of more network services without exceeding network limitations, including the implementation of security measures.

4.1.1 Chapter Layout

This chapter is presented as follows:

- Section 4.2 defines terminology used throughout the chapter.
- Section 4.3 outlines the research methodology used to conduct the analysis of CBBA, proposing two novel algorithms to address the communication complexity of consensus-based DTA on MANETs.

- Section 4.4 identifies specific process modifications and variables to allow quantitative observations of CBBA and the proposed algorithms.
- Section 4.5 introduces Cluster Form CBBA (CF-CBBA), a novel algorithm that aims to reduce communication complexity by dividing the network into clusters while still arriving at a global solution to task allocation problems.
- Section 4.6 introduces Broadcast Enabled CF-CBBA (BECF-CBBA), a novel modification of the way in which bundles and state information are communicated during task allocation.
- Section 4.7 summarises the chapter.

4.2 Terminology

Several terms form the core of this investigation, describing vital attributes and phenomena which have been deemed vital for the research in this chapter. These terms are:

- Communication event; a one-way exchange of data between a source and destination node. Assumed to be a single packet unless otherwise stated.
- Network resources; the number of bytes required to communicate during DTA, to provide a solution to the initial problem.
- Solution; a full CBBA task allocation process. This may involve multiple rounds.
- Round; one full exchange of bundles between all member nodes.
- Bundle; data representing the current state of a node. A bundle includes tasks, current bids and a list of which tasks the source node has selected in order of optimality.

4.3 Methodology

To address the issue of control overhead in autonomous MANETs, brought about by the use of CBBA, this chapter discusses the communication and computation requirements

of CBBA and proposes two approaches to optimising the communication required to reach consensus.

Control overhead is a term used to describe the number of transmissions and the size of the packets that comprise those transmissions. It represents the sum cost of communication for control functions. It differs from network overhead, but being derived from application layer processes that require communication but play no part in the management of the network itself. Network overheads include routing and security services, whereas control overhead relates to communication required by CBBA, VoIP and other application layer programs and protocols. A preliminary investigation of the communication requirements of CBBA will be conducted. This analysis informs the proposal process by providing quantitative data regarding the complexity of communication under CBBA using currently proposed methods of task allocation over wireless networks. The primary areas of investigation are:

1. The effects of problem domain and network size on the optimality of an assignment.
2. The number of communication events required to achieve a solution.
3. The number of bytes transmitted during DTA.

These areas of investigation are derived from two hypotheses outlined in Chapter 3, Sub-section 3.4.2:

- *Consensus-Based Bundle Algorithm's cost in number of transmissions and bytes communicated can be lowered by dividing the MANET into clusters.*
- *Multicast communication, with additional communication to facilitate CBBA's synchronisation over the network, will reduce the number of transmissions and bytes required by DTA.*

Reviewed literature has shown that DTA can be communication intensive, either in raw communication requirements or in the amount of relayed traffic. Investigation 1 is required to analyse the root of this communication complexity; identifying the effects of increasingly complex missions and network sizes on both the optimality of the

assignment. This will allow the means by which optimal mission scores may be achieved to be identified and quantified in terms of a ratio of nodes and tasks.

Similarly, Investigation 2 will allow the quantification of communication events, the number of transmissions and retransmissions required to achieve a convergent state in the network. The effects of increased network size and mission complexity will be analysed to provide information regarding what attributes of the problem domain and network contribute to communication complexity. This will allow identification of potential methods by which communication complexity may be reduced.

Chapter 3, Section 3.2, identifies issues regarding the utilisation of network resources for the completion of DTA processes. Nodes that are required to repeatedly send large bundles of tasks represent a significant network resource cost. Such expenditure of resources is wasteful in the context of this research, and may represent a significant problem for resource constrained mobile networks. Investigation 3 will allow the number of bytes required to communicate a solution to be identified, providing a quantitative means of analysing the problem. Such analysis will be used to identify mitigation measures that may reduce the network resource cost of performing DTA, reducing incidental waste of network resources.

The deconstruction of the two hypotheses into testable elements based on three key areas of investigation leads to the following proposed simulations:

- Communication events over different network and problem sizes.
- Network resources to complete DTA over different network and problem sizes.
- Mission score over different network and problem sizes.

These simulations will be performed in MATLAB, using CBBA as a basis for comparison with any proposed methods. In the preliminary analysis undertaken in this chapter, CBBA will be simulated under the conditions outlined in Table 4-1 to identify trends on the baseline algorithm that may provide a foundation for the proposal of solutions to mitigate the cost of DTA on network resources.

Table 4-1 Table showing the critical variables for preliminary analysis of CBBA

Attribute	Value
Number of nodes:	6, 9 and 12
Number of tasks:	1 to 50
Size of Mission area:	100m by 100m
Random number generator:	MATLAB “rand”
Seed value:	11
Number of Iterations:	100

A range of network sizes are to be investigated. The smallest network will contain 6 nodes. Initial investigations showed that 3 node networks had very simple communication, but would achieve very low mission scores due to the low number of nodes to tasks in more complex missions. As a result, 3 node networks were found to be unsuitable for missions with more than 15 tasks. Networks of 6 nodes provided more complex communication, but were able to arrive at solutions with above 40% optimality, making them the smallest networks selected for comparison.

The maximum size of network tested contains 12 nodes. This is a conservative limit set by the size of the mission area to avoid node collision, as outlined in the CAP 722 (Haddon & Whittaker 2003). This document outlines that UAVs may not pass within 30 meters of unaffiliated infrastructure, objects or people. The Global Positioning System (GPS) provides an approximately 8 meter range accuracy with 95% confidence. Up to 18 nodes may occupy such an area while observing this limitation, but for the purposes of this initial analysis, the three network sizes chosen are considered sufficient to highlight the effects of network size on communication events and assignment optimality, based on previous research conducted by (Brunet et al. 2008).

To ensure that nodes have sufficient room to navigate the mission area with contravening the hard limit set by the CAP 722, and to avoid collision with member nodes of the network (relying only on GPS data for anti-collision measures), a maximum of 12 nodes is allowed in a 100 square meter mission area. These factors have been taken into account to reflect a likely and realistic set of parameters that must be met in real world UAV

deployment scenarios. 9 node networks have been selected to provide a mid-point between these minimum and maximum sizes.

A range of problem domain sizes, ranging from 1 to 50 tasks, has been selected. The smallest problem domain sizes will result in redundancy of nodes, due to fewer tasks than nodes being present, but will still facilitate the identification of the communication complexity, bytes transmitted by the whole network, and optimality of such assignments. Larger problem domains, up to 50 tasks, may begin to over-subscribe individual nodes with tasks. DMG and the cost of travelling to tasks will greatly depreciate the value of tasks placed 5^{th} or lower in bundles held by individual nodes, potentially reducing mission score (and thus optimality of the assignment) substantially. As a result, 50 task problems are the largest selected for simulation when analysing CBBA.

A mission area of 100 square meters has been selected, to provide a simple space within which tasks and nodes may be randomly placed to facilitate the simulation of DTA. Nodes remain stationary during the task allocation process, so as to preserve their initial positions and thus the validity of bids between rounds of communication. The MATLAB rand function, using a seed value of 11, is used to randomise each iteration of the experiment.

The experiment is run for 100 iterations, to provide trends with a high confidence value, mitigating the irregularities that may occur in instance where the network or tasks may be distributed poorly (clumped in one area). Task scores are set to a default value of 100 per task before associated costs and DMG.

The outcomes of these simulations will be used to provide an initial analysis of CBBA, and provide a foundation for discussing potential mitigation measures. These include potential modifications to the means by which nodes distribute and communicate tasks, as outlined in the introduction of this chapter.

The platform-specific constraints and capabilities of a node will affect some outcomes of these experiments. The cost of remaining airborne, for example, is platform specific. The next sub-section will discuss platform specific considerations to provide a description of the target platform considered when analysing the preliminary experiments that will be undertaken in this chapter.

4.3.1 Platform-specific Constraints

Mobile nodes take a wide range of forms, each with unique characteristics. The most common fundamental costs of a mobile node, common to all such nodes, are:

- Propulsion.
    - The energy cost of movement and in the case of UAVs a constant cost associated with remaining airborne.
- Payload.
    - Computer hardware and sensory equipment will use energy. The rate of use is variable, but there is a baseline cost associated with such items.
- Communication.
    - The hardware that enables wireless communication also requires power, with baseline costs and transmission costs.

This research considers lightweight, resource constrained UAVs, such as quadrotors as an example of platform-related cost. Propulsion is the most demanding aspect of a UAV, especially in the case of quadrotors. To remain airborne, such platforms must continually spin their propellers, drawing power from the battery constantly. The rate of consumption will vary based on the altitude desired and any corrections to course required, as the spin rate of the motors will be adjusted to provide the required movement, but a baseline rate of power consumption can be calculated based on a node hovering at a fixed altitude.

Payloads also draw power constantly, but at a much lower rate than propulsion. The rate of consumption will vary on the type and amount of computational hardware installed on the platform, but will generally remain low when compared with the costs associated with propulsion.

Communication also requires power, but even less than the hardware installed on the platform. The baseline cost of keeping a radio on is usually associated with the hardware cost of the platform, with transmission being an independent cost associated with every packet sent by the network interface through the transmitter. This cost will increase as communication becomes more frequent, but will remain low compared to the cost of hardware over the course of a mission.

Propulsion being the highest cost in a quadrotor UAV has some serious implications for the way in which DTA is performed. The closing paragraph of Sub-section 4.3.3 states that CBBA requires that nodes remain stationary or assume their initial position is constant to prevent race conditions when assigning tasks. If nodes are required to remain stationary, the cost of communication as time taken to communicate may also incur a high cost in energy as constant power must be supplied to keep the platform in the air. It cannot be assumed that the platform can land to save power, as it is important to limit the possibility of physical access to the node for security purposes, and it may not always be feasible to do so. Landing reduces the effective range of wireless communication and terrain may be unsuitable for a landing (sea rescue operations being an example of such an environment). The measurement of this cost is considered to be outside of the scope of this research, but as a means of highlighting the potential real world costs of overly complex communication on the ability of a network to complete tasks effectively with limited energy resources, it is a useful idea to consider.

Timely communication will provide benefits in terms of the rate at which tasks may be undertaken and missions completed, which will reduce wasted power when considering the energy cost of keeping a platform active in the mission area. Timely communication, in the context of this research, may be described as communication that falls within the network resource constraints of the target network, with a minimum of communication redundancy and a high resultant optimality of the DTA procedure.

4.4 Preliminary Analysis

Chapter 3 provided an overview of the attributes and constraints of CBBA. This sub-section focusses on MANET-specific implementations of CBBA to provide a baseline for further investigation of methods to reduce communication complexity and the network resources required to solve task allocation problems. The focus of this sub-section is the communication cost of problem solving using CBBA. The algorithm itself is not modified in this chapter, the focus being on investigating methods of implementing the baseline algorithm in novel ways which provide benefits to MANETs comprised of lightweight, limited-resource nodes. Platform-specific constraints play a role in the ability of nodes to

participate in CBBA, and these constraints are analysed to contextualise observations made regarding the requirements of CBBA and the ability of a MANET to meet them.

4.4.1 Operational Characteristics

CBBA is essentially a means of allocating resources to simple elements of complex problems, thereby solving the larger problem by addressing it piece by piece. A simple scoring mechanism is used to determine the fitness of a node for a given task, calculating the cost of a task and placing a node specific value (bid) if the value is positive.

Preliminary analysis of CBBA found that the optimality of an assignment will decrease as the number of tasks becomes too large for nodes to service in a reasonable timeframe. An artificial value, Diminishing Marginal Gain (DMG), is applied by CBBA to ensure that tasks lower in a bundle are never worth more than those placed higher, and this causes an incremental drop off in reward the lower a task is rated on one node.

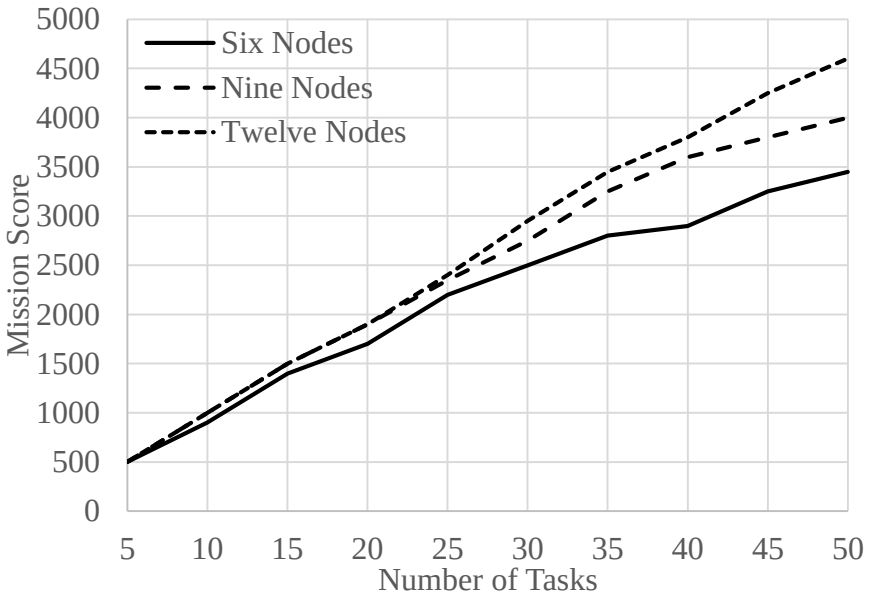

Figure 4-1 Graph comparing the effects of different sized networks on assignment optimality

Figure 4-1 shows that to maintain the optimality of an assignment, the number of nodes must be sufficient to ensure that bundles do not become overlong. In small networks

required to assign many tasks, bundles become large and the effects of DMG become apparent quickly. The characteristics of the simulation used to conduct this preliminary analysis are detailed in Section 4.3.

Optimality is calculated by comparing the achieved score to the total potential score. Two associated costs reduce mission score; the energy spent to get to the task (usually an abstraction, such as -1 point per meter – this is always implementation specific), and DMG. DMG is applied to tasks when they are not the first task in that bundle. For every place lower than first, a small subtraction is made, to enforce and order of execution on tasks that are close in terms of potential score.

This ensures that bids and score lists reflect a basic concept of opportunity cost; if a node takes on two similarly scored tasks, and another node puts in an equal bid on the second of those tasks with no other active tasks in its bundle, the contesting node will win due to DMG reducing the value of the first nodes bid. As a result, both tasks are executed in parallel when the nodes are sent to execute their tasks lists, instead of one node executing two similarly scored tasks sequentially.

With minimum cost of travel between nodes and tasks (provided by globally optimal solutions to tasks), six node networks only achieve scores of 3500 out of 5000 (70% optimality) compared to 4612 out of 5000 (92% optimality) for twelve node networks when performing DTA on 50 tasks in each network configuration. This is reflective of DMG deprecating the value of tasks allocated to the bottom of the bundles held by over-allocated nodes, as well as the costs associated with servicing so many tasks (some of which may cause their executing nodes to move away from other bundled tasks, raising the cost to execute them at run time and after allocation). With more nodes, more tasks may be allocated to nodes with only a small number of existing tasks in their bundles, spreading the workload and increasing efficiency.

As network size affects the complexity of communication during task assignment, this represents a constraint of CBBA when considering potentially large problem domains.

4.4.2 Communication Complexity

CBBA requires that all nodes communicate their bundles in rounds, with all other nodes to arrive at a globally optimal solution after a number of such rounds. It has been identified that communication stability, Quality of Service (QoS) and speed have not been investigated in depth, as they are outside the scope of this research. CBBA and its extensions are control algorithms and much of the research related to them is focused on the topics of control and computational complexity.

Communication complexity in CBBA is represented by the total number of communication events required to drive a network to consensus. The higher the number of events, the higher the complexity of communication. This may be directly mapped to the network resources required to communicate a solution; increasing communication complexity will require more bytes to be sent. Contributors to communication complexity under CBBA are:

- The number of participating nodes.
    - This is generally assumed to involve the whole network, but this is not always the case. CBBA extensions such as Team CBBA, proposed by (Hunt et al. 2012), support the compartmentalisation of a network into individual groups for the purpose of task allocation.
- The number of tasks.
    - In exceedingly large problems, the task list may require multiple packets to be shared between nodes. However, in missions with less than 200 tasks with tasks having X, Y, Z coordinates, a unique ID, a winning bid ID and value as attributes, it is unlikely that the MTU of the interface will be exceeded. This means that only one packet will be sent to transfer a bundle, limiting the impact of the number of tasks on the number of transmissions required to reach convergence, in scenarios requiring less than 200 simple tasks.
- Granularity of bundle sharing.
    - Bundle sharing in CBBA is sequential, as nodes may not update their bundles until they have state information from other nodes to compare

against their local assignments. To prevent data collision on the network, nodes take turns to communicate with each other. The manner in which this is done (unicast, multicast or broadcast) will have an effect on the number of communication events required.

- Number of rounds.
    - The number of rounds required to complete a CBBA allocation is determined by the number of participating nodes and tasks. CBBA requires at least two rounds to complete, as a verification round in which bundles do not undergo modification is required in addition to the number of rounds required to arrive at a solution.

The number of rounds is highly variable and difficult to predict, though preliminary analysis of CBBA has found that it does not exceed 5 rounds on networks up to 20 nodes in size and in problem domains of up to 100 tasks. The number of rounds does not increase with the size of the problem and the network, instead increasing when the division of tasks among nodes becomes contentious, with several nodes having closely matching or tied bids on a small number of tasks. Additional tie-breaker rounds are required to resolve such issues.

Unicast transmission is used, for the purpose of this analysis. CBBA is connection-oriented, it requires that nodes receive bundles cleanly, without collision on the channel. To ensure that the channel is clear, nodes take turns to communicate and each node addresses every other node in the network sequentially, so that acknowledgements do not collide.

Figure 4-2 demonstrates the output of a preliminary simulation of CBBA under the conditions outlined in Section 4.3, showing that the number of nodes in a network has a clear impact on the communication complexity of the network. Networks comprised of twelve nodes require 9.5 times more communication events to reach consensus than six node networks. The irregularity of the lines is caused by the variable number of rounds required to reach consensus.

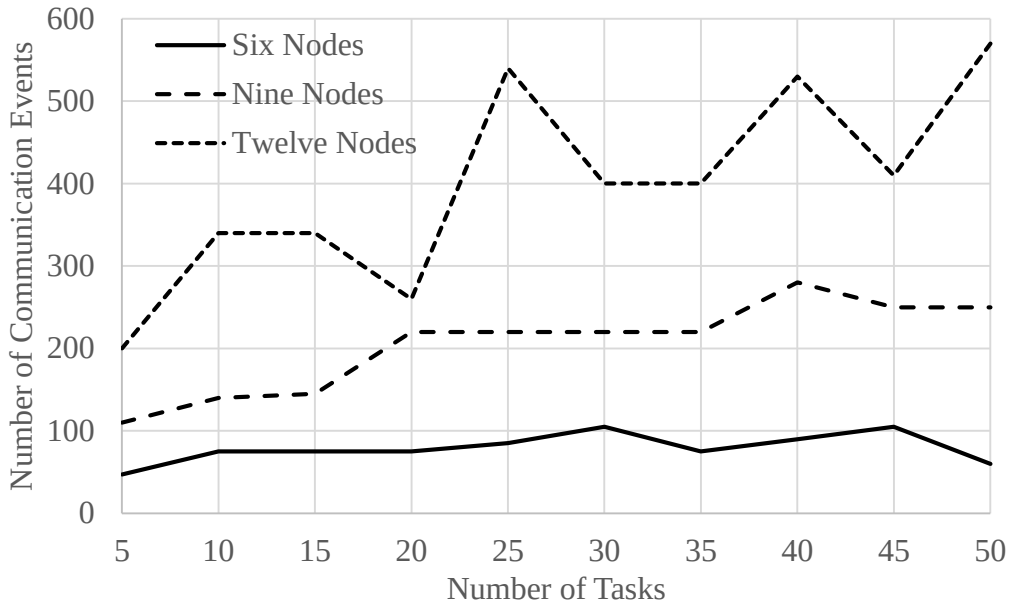

Figure 4-2 Graph comparing the effects of three sizes of network on communication complexity

To successfully arrive at a solution, nodes undergoing CBBA must remain stationary or assume that their initial starting point is used for the entire DTA process. If nodes recalculated their bids based on their updated position each round, the results would change and the process might enter a race condition, in which the mobility of the nodes drives them away from consensus due to changing conditions.

4.4.3 Section Summary

CBBA's communication complexity has been analysed and found to be highly dependent on the number of nodes in a network. The number of tasks has a lesser effect, by increasing the chances of more rounds being required when allocating large numbers of tasks.

Due to the probabilistic nature of wireless communication, the more individual communication events occur, the more likely it is that one will become corrupted or lost. Lost data must be retransmitted, increasing the complexity of communication further. The dynamic topology of MANETs can further complicate matters, requiring event-driven or periodic updates of the routes between nodes depending on the routing protocol. Routing

services and security also make use of the communication channel, so knowing the communication requirements of DTA allows the utilisation of the channel to be predicted. This can assist in determining if a given approach is suitable for the network resources available to a given MANET.

Two core issues have been identified that will be addressed by the proposals in this chapter: the need to reduce communication complexity (to reduce the chance of data loss and reduce utilisation of the channel to avoid impairing other network services), and the need for efficient DTA communication to reduce wastage of network resources.

4.5 CF-CBBA: Topology Inspired Optimisation of CBBA

Communication complexity under CBBA increases with the number of nodes involved in task assignment and the number of rounds required to reach convergence. The size of the network has an effect on the number of rounds, as previously identified, further increasing the effect of the number of nodes on the amount of communication required to achieve consensus.

By reducing the number of nodes involved in task allocation, the complexity of communication may be reduced. However, it has also been identified that to retain optimality in larger problem domains, the network must be of sufficient size to allocate all tasks in a manner that reduces the impact of DMG scoring on the resulting assignment for each node.

The conflicting requirements of optimality and reduced communication complexity may be addressed by forming temporary clusters of nodes. Due to the node-based nature of MANETs, wherein all nodes form both the network infrastructure and end-points for communication, it is possible to dynamically assign nodes to clusters based on a variety of factors. Route length between nodes may be used to determine groupings that require less communication over intermediate nodes, reducing additional traffic generated when intermediate nodes relay bundles during task allocation. Physical positioning may be used to determine nodes that are likely to share a set of tasks which the rest of the network are unlikely to be found suitable for.

Regardless of the selected method of clustering, it is possible to sub-divide the network into smaller groups of nodes, reducing the communication complexity of the task assignment process on each cluster.

4.5.1 Forming a Clustered MANET for DTA

The objective of CBBA is to provide a globally optimal solution for the network. If a number of clusters are operating independently, this condition cannot be met reliably, due to the lack of communication between clusters. It is therefore not possible to retain the task allocation functionality of CBBA in a network that arbitrarily divides a list of tasks between clusters.

The network clustering process must adhere to the following requirements:

- The resulting allocation must be globally optimal.
- Clusters must have a means of cross-cluster communication.
- Division of the problem domain cannot be arbitrary, an initial assignment stage is required to provide task lists suitable for individual clusters.

To ensure that the result of convergence is a globally optimal solution, cross-cluster communication and non-arbitrary division of labour must be observed. Cross-cluster communication and the initial division of labour may be viewed as a single problem; there must be a control element that was not previously required by CBBA that determines which clusters will work on what tasks. This can be achieved by the division of nodes into two sub-classes; cluster heads and member-nodes.

Figure 4-3 shows an unclustered network. The clustering process has not begun, and task allocation cannot yet be undertaken. Single lines represent logical links, or direct lines of communication, between nodes.

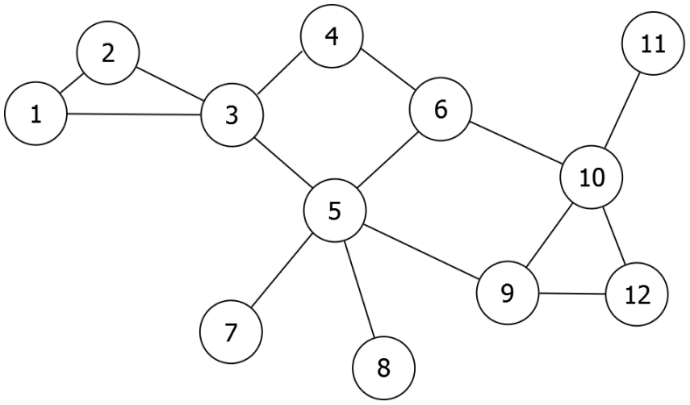

Figure 4-3 Unclustered MANET (single lines represent logical links between nodes)

Figure 4-4 shows the network after cluster-heads have been selected. The red nodes have been selected as cluster-heads due to their combination of high connectivity with other nodes and the proportion of routes which are no more than one hop in length.

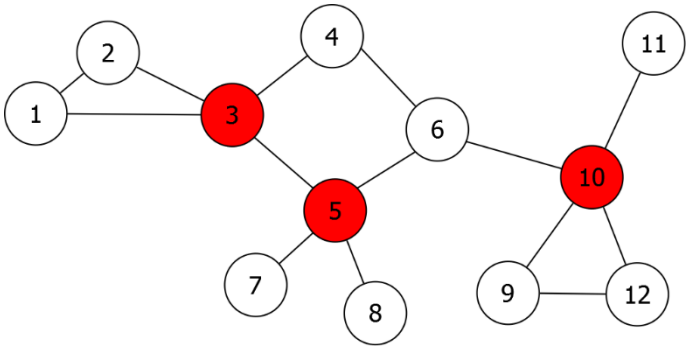

Figure 4-4 Cluster-heads selected, MANET still unclustered

Cluster-heads are assigned control of the initial task allocation process, performing CBBA to produce bundles for each cluster-head in the network. Only cluster-heads participate in this stage of the allocation process. Member-nodes perform network-related services during this stage of task allocation, relaying messages between cluster-heads and participating in routing operations.

Figure 4-5 displays the final result of the clustering process. These clusters are DTA-specific, they do not dictate the normal behaviour of routing protocols. Only task allocation processes are affected by the designation of clusters, forming a list of cluster-

members at the application layer to inform nodes as to who their neighbours are when performing CBBA.

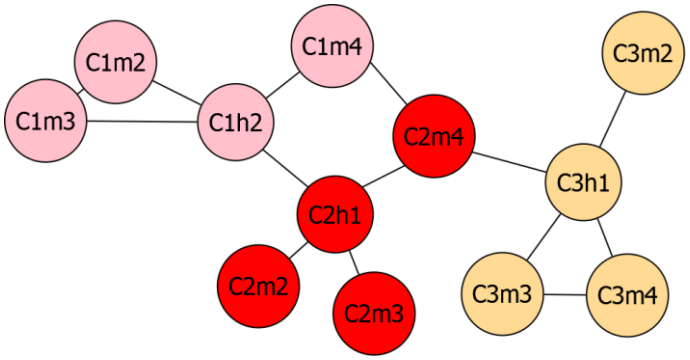

Figure 4-5 Clustered MANET (single lines represent logical links between nodes)

Member-nodes perform task allocation within their local cluster. The initial allocation process provides each cluster with a sub-set of tasks derived from the initial task list, providing the basis for further sub-division of the problem domain between all members of the cluster. The cluster-head is considered to be a member-node at this stage, having no special status but to be the first node to communicate its bundle to other members of the cluster.

4.5.2 Assigning Tasks to Clustered Nodes

Having defined the means by which clusters are formed, the task allocation process must be defined. The process must utilise all nodes in the network and provide a globally optimal solution.

CBBA provides the baseline algorithm for CF-CBBA (Smith et al. 2014). When considering the individual task allocation process, CF-CBBA does not algorithmically differ from CBBA.

An initial task allocation process is required to divide the problem domain between all cluster-heads. Figure 4-6 shows the nodes involved in an example cluster-head allocation process.

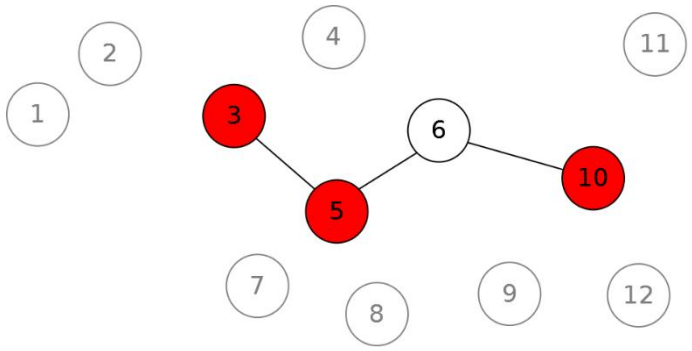

Figure 4-6 Diagram showing cluster-head communication of task bundles

The red nodes are cluster-heads and participate directly in this stage of task allocation. Greyed-out nodes are non-participants with no responsibility for routing under the current topology. The white node (6) is a member-node that is acting as a router between one or more cluster-heads, while not participating directly in DTA communication.

The network must retain the ability to route over non-participating nodes, as cluster-heads may not be in range. The lines, representing logical links between nodes, show that node 6, a member-node, may route communication between nodes 3 and 5 to node 10. This reinforces the earlier assertion that CF-CBBA only partitions the higher-layers of the network to preserve the routing capabilities of the whole MANET.

CBBA is performed between the cluster-heads until consensus is reached regarding the initial division of the problem domain into three local bundles.

Cluster-heads could use the average of the positions of all nodes in their cluster to submit bids, to represent the fitness of the cluster for a given task, instead of the cluster-head itself. This is a measure intended to reduce the potential for sub-optimal allocation of tasks, should the position of nodes in the cluster differ greatly from that of the cluster-head. It is the implementation that determines if this method of cluster allocation is used, though. Different implementations may have different requirements, depending on the

heterogeneity of nodes and the attributes required of individual cluster (such as clustering by capability, or location). As such, it is most often the implementation that will determine the means by which a cluster head will perform bids on behalf of their cluster.

A second phase of task allocation is required to distribute the resulting cluster-head bundles between the member-nodes of the clusters. Figure 4-7 shows the three previously defined clusters, with cluster-heads assuming the role of member-nodes for the purposes of DTA.

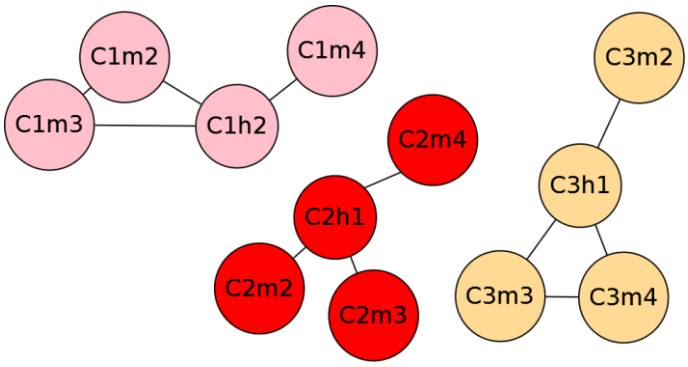

Figure 4-7 Diagram showing cluster-member communication of task bundles

Each cluster performs task allocation in isolation from the others. The task bundles computed by cluster-heads are distributed among members of that head's cluster. CBBA is then performed at the cluster-level with no cross-cluster communication of tasks between clusters, until a solution is found for each cluster. It remains possible for nodes to route over other clusters during this phase of CF-CBBA and all non-DTA network services are provided across the whole network.

Figure 4-8 provides a flowchart of the task-allocation process. In addition to the division of the network into clusters shown above, the flowchart demonstrates the DTA process.

DTA is described as a two-loop process under CF-CBBA. The outer-loop, or cluster-head allocation stage, performs the role of problem domain partitioning. Its initial task list is communicated to all cluster-heads, who proceed to perform CBBA until convergence is reached.

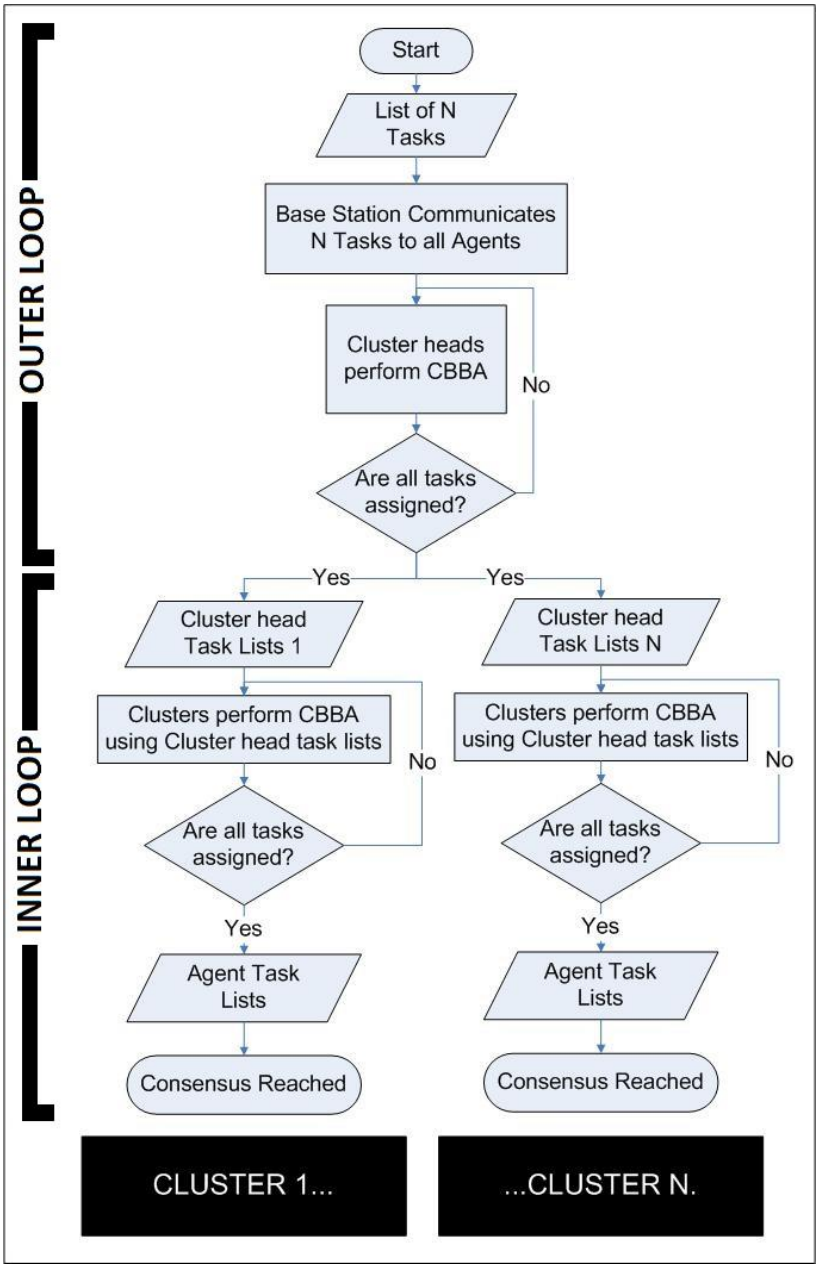

Figure 4-8 Flowchart showing the CF-CBBA task allocation process

The resulting bundles are not committed to an active list of tasks to be executed by the node, but are stored as a new task allocation problem. This problem will be further sub-divided among the members of the cluster in the inner-loop of the process.

The inner-loop is a parallel process, with each cluster performing its own CBBA process independently. The clusters use the results of the outer-loop allocation phase, shared with them by their cluster-head, as a new task list. Upon reaching consensus, the resulting task

bundles are then committed as the local executable task list for each node, which may be acted upon immediately, independent of any clusters which may take longer to complete the DTA process.

As previously discussed, the communication events required to reach consensus under CBBA are proportional to the number of rounds and the number of participating nodes. CF-CBBA's use of a two-stage allocation process requires that two instances of CBBA are run sequentially, the initial cluster-head allocation and a parallel allocation phase across all clusters.

Due to the unicast communication model used for this simulation, each node must communicate with every other node. This leads to the complexity of a single rounds worth of communication being expressed as *n.(n-1).* Equation 3-1 on page 46 provides the full expression allowing the calculation of the total number of transmissions required in a fully connected network to complete a full CBBA process. The maximum observed number of rounds required for a 12 node network undertaking a problem of 50 tasks, is 5 rounds. This includes 4 allocation phases and 1 validation phase.

Equation 4-1 demonstrates the communication complexity of CF-CBBA (Smith et al. 2014). This equation is a means of calculating the total transmissions required using unicast communication, with T_i and P representing instances of Equation 3-1, originally defined by (Johnson et al. 2010).

$$\boldsymbol{E = \left(\sum_{1 \leq i \leq C} T_i \right) + P} \qquad \textbf{(Equation 4-1)}$$

T represents an instance of cluster-level (inner-loop) task allocation, with i providing the instance identifier. The sum of all cluster-level communication events, plus the result of the initial cluster-head allocation represented by P, results in the variable E that represents the total number of communication events required by CF-CBBA. T_i and P are calculated using Equation 3-1, and are equivalent to the result, x, where all other variables match the conditions of the instance of task allocation being modelled.

CF-CBBA is designed to reduce the number of nodes involved in a given stage of task allocation, thereby reducing the total number of communication events compared to baseline CBBA for an equivalent number of nodes. *P*, being a non-parallel task allocation process, is added to the sum of all *T*. Each instance (*i*) of *T* represents the total communication events required to reach consensus within a cluster. This means that each *P* and *i* value is calculated using Equation 3-1 when considering unicast communication.

The number of bytes sent is proportional to the number of communication events and the size of the bundles being transmitted. A reduction in either variable will reduce the network resource requirements of the DTA process. By clustering the number of communication events may be reduced as discussed with relation to Equation 4-1. An additional benefit may be observed in the division of the problem domain into smaller chunks for processing at the cluster level. As the cluster heads submit their bundle assignments as new problems for their clusters, these new problems are smaller than the original list of tasks. This reduces the number of tasks in each bundle at this stage of allocation, further reducing the network resource requirements at the cluster level. These benefits will not be seen at the cluster head level, as they must allocate the entire problem domain between themselves, gaining no benefit of problem sub-division until the cluster stage.

The validity of the above statements will be proven in Chapter 5, which will provide a test plan and investigation of the communication characteristics and optimality of CF-CBBA compared to baseline CBBA. There are some communication methods, considered out of scope for this research, which allow the avoidance of data collision between clusters using different channels for each cluster. It is assumed during simulation that there is a zero probability of collision due to such methods existing, but their analysis is considered to be outside the scope of this research.

4.5.3 Variable Cluster Sizes

The previous sub-sections in this chapter assume that clusters are comprised of equally sized clusters. This assumption is carried forwards into the simulation stage, but should be discussed further to highlight key issues and potential solutions.

At the cluster-head allocation stage, the assumption that all clusters are of equal size will have no effect, as a network of 3 clusters of 3 nodes each, and a network of 3 clusters with 2, 4 and 3 sized clusters will both have three cluster heads. However, when the tasks are allocated at the cluster level, there may be issues caused by potentially equal division of labour.

The cluster head CBBA stage may result in similar sized task-lists on clusters that are not equal in size. This will be affected by the relative positions of the clusters within the mission area, but assuming that all variables are equal across clusters, except for cluster size, the issue of node over/under subscription remains. The potential result of task allocation in such circumstances includes a loss of optimality due to nodes taking on more tasks than would be optimal, incurring the cost of DMG as discussed in Chapter 3. Nodes in larger clusters may end up taking on far fewer tasks than they otherwise would be able to.

A potential solution to this would be the implementation of a fitting function that accounts not only for the relative position of the cluster within the mission area, but the number of nodes in each cluster. Such a function could also be extended to provide contextual filtering of available nodes, for example nodes with a payload suitable for a given task will be given preference over those able to perform the task, but not specifically equipped for it.

The implementation of such a fitting function is considered to be outside of the scope of the research, but could be an item of future work. Nodes incapable of performing tasks are already filtered out of CBBA, CF-CBBA and BECF-CBBA by using task-type flags in the node description to highlight tasks which the node is capable of performing.

4.5.4 Section Summary

A novel method of reducing the communication complexity of CBBA has been proposed. This algorithm, CF-CBBA, requires that the network be divided into clusters to reduce the number of nodes in each allocation phase. The number of tasks at the cluster-level parallel allocation phase is reduced due to the initial cluster-head allocation phase having divided with initial task list into a number of bundles equal to the number of cluster-heads.

As the number of communication events required by CBBA is proportional to the number of nodes multiplied by the number of rounds required to converge to a solution, reducing the number of nodes at each allocation stage is proposed as a means of reduction of that complexity. Further reduction of communication complexity may be achieved through the reduction of the total number of rounds required, which is itself driven by the combination of the number of nodes and tasks. It is proposed that by reducing the number of tasks and nodes, the number of rounds required for an allocation may also be reduced, further reducing communication complexity.

4.6 BECF-CBBA: Investigating Wireless Communication

Wireless networks commonly use omnidirectional radio transmitters to send messages. The propagation of the transmission in all directions makes it an inherently broadcast medium; one cannot control the spread of the transmission, only its range. While presenting security risks when considering ease of observation and derivation of network-related information, the inherently broadcast nature of wireless communication may be of benefit when considering methods of reducing the communication complexity of CBBA and CF-CBBA.

4.6.1 Preliminary Analysis of Wireless Communication for CBBA

Consensus-based DTA algorithms, such as CBBA, require that all participating nodes are able to share bundles, updating the network with local state information from each node in every round of allocation. Wireless communication can take one of three forms: unicast, broadcast and multicast. The control of communication is defined by which of these methods is used.

4.6.1.1 Unicast Communication of Bundles

Unicast transmission involves end-to-end communication of messages. This method allows a great deal of control over where bundles are sent during task allocation, allowing individual nodes to be addressed.

The unicast communication of task bundles can provide:

- Targeted communication.
    - As nodes communicate to a single destination per bundle-transmission, messages may be node-specific.
    - This is of benefit in heterogeneous networks where nodes may have unique characteristics that make them more suitable for some tasks and completely unsuitable for others.
    - By addressing nodes individually, the source node may have more control over what is sent, cutting down on redundant communication.
- Guarantees of delivery through acknowledgement of packet reception from the destination node on a one-to-one basis.
    - By acknowledging that a bundle has been received, control of communication flow in the network may be enforced.
    - It has been shown that communication under CBBA is complex and that orchestration of communication is required to prevent packet collision or nodes falling out of sync and communicating over each other.

    - By acknowledging successful delivery in an inherently probabilistic medium, reliability may be improved.
- Synchronisation of node and network state information.
    - State updates are the core of CBBA functionality.
    - By using unicast communication, the change in state information can be controlled by targeting nodes for updates in a selective or ordered manner.
    - This can reduce duplication of node-specific data.

Unicast communication is a verbose form of communication. It requires that a unique packet be sent for each node-to-node communication event. The communication cost of unicast, in terms of communication events, is expressed by Equation 3-1 when considering CBBA communication requirements. The additional control of information flow offered by unicast communication comes at a significant cost in communication complexity.

4.6.1.2 Broadcast Communication of Bundles

As radio transmissions are inherently broadcast, it seems logical to utilise this characteristic to reduce repetition of communication. Unicast communication of CBBA incurs a high-degree of packet redundancy, as a single node in a homogenous network will send the same bundle to every destination node in the network. By broadcasting such information, redundancy may be reduced by only requiring a number of transmissions required to reach all nodes in the network.

In fully connected networks, in which all nodes are in range of all other nodes, a single broadcast may be sufficient. However, it is unlikely that a MANET will remain in such a configuration for the whole mission, requiring a means of ensuring that broadcast bundles are propagated throughout the network.

Network flooding is an expensive, but effective, means of propagating data throughout the network. CBBA with Relays also provides a method of designating nodes as relay nodes, which ensure that bundles are propagated throughout the network. The following must be considered when implementing a broadcast bundle-sharing mechanism:

- Adequate channel-control must be maintained.
    - Collision of acknowledgements or relayed packets must be prevented; Carrier-Sense Multiple Access (CSMA) is an example of a means by which nodes may time communication so as not to overlap on the channel.
- The synchronisation of nodes becomes explicit, due to the lack of communication-driven synchronisation.
    - The order in which nodes transmit bundles will have to be explicitly communicated so that the sequential element of CBBA, the update of global state information, may function correctly.
    - The order itself is not important, only that nodes do not simultaneously attempt to send bundles.
    - Unicast communication circumvents this issue by addressing nodes individually, allowing control of the order of transmission by ensuring that one destination will always be the first to compute its local bundle (as it is the first to receive the previous node's bundle).
    - Control communication that reinforces this order will be required under broadcast communication.
- Under CF-CBBA, broadcast packets may represent redundant communication when received by neighbouring but unrelated clusters of nodes.
    - CF-CBBA inner-loop task allocation requires that clusters work on nodes independently, while retaining all network services unrelated to DTA.
    - Sub-netting the network is not a viable solution, as this would require the use of gateways to allow communication across clusters, adding further complexity to communication in the form of control traffic.
    - As the purpose of broadcast communication of bundles is to reduce communication redundancy, the reception of out-of-cluster bundles is deemed undesirable.

4.6.1.3 Multicast Communication of Bundles

Multicast communication lies between the fine control of unicast and the far-reaching single transmissions of broadcast communication. By designating a sub-set of nodes, communication may occur selectively, but across a number of nodes instead of being locked to a single address.

This may be of benefit to CF-CBBA, as clustered nodes have knowledge of the address-space of their cluster, allowing controlled transmission of information between nodes without wasteful reception and processing by other clusters. This may also allow cluster-heads to communicate more efficiently.

Multicast communication provides the control benefits of unicast, while reducing the number of redundant transmissions by allowing a limited form of broadcast within the range of addresses described. It does, however, encounter the same issue as broadcast when considering the control of information flow over routes longer than one hop. Should a cluster become dispersed throughout the network, the retransmission of multicast messages is a vital consideration.

4.6.2 BECF-CBBA: Multicast and Broadcast communication for Clustered DTA

BECF-CBBA is proposed as a means of providing the control required to relay multicast and broadcast communication through a MANET. The aim of this proposal is to address the issues of redundant communication while maintaining the complexity reduction measures of CF-CBBA.

Communication is divided into two categories; global and incremental state updates. Global updates represent communication that must reach all nodes in the network, or which benefits from local broadcasts. Incremental updates require the use of multicast communication to control the flow of information and communicate it only to those nodes that need to be involved in that communication.

Table 4-2 shows the mapping of BECF-CBBA communication types to these categories. Bundle sharing is the most common form of communication under a consensus-based algorithm, as the sharing of information between nodes is vital for the convergence to a state of consensus in the network. Under BECF-CBBA this is an incremental update, occurring first at between cluster-heads, then the members of clusters. Due to the multicast nature of incremental updates; node selection, round and allocation completion messages are required to ensure that nodes are informed of the current stage of task allocation.

Table 4-2 CF-CBBA message types mapped to Global/Incremental categories

BECF-CBBA Messages	Global	Incremental
Bundle sharing		☑
Cluster-head nomination	☑	
Cluster-forming	☑	
Allocation completion flags		☑
Round completion flags		☑
Next node selection		☑

Network control communication required to divide the MANET into cluster-heads and clusters is global. These updates require that all nodes are made aware of their designations, any change in their role within the network and what groups they will perform inner-loop CF-CBBA with.

To ensure that all nodes in the network receive broadcast messages, it would be appropriate to use network flooding. The impact of flooding on the number of communication events is controlled by only broadcasting global updates, reducing the amount of flooding and ensuring that only simple messages are flooded to prevent saturation of the communication channel.

Multicast communication is more common under BECF-CBBA. Incremental updates occur frequently, but require no special handling of out of range nodes. MANET routing

protocols such as OLSR and AODV handle multi-hop multicast by forming routes for each individual destination node, allowing the propagation of transmissions from source to multiple destinations without special considerations above the network layer.

4.6.3 Section Summary

BECF-CBBA is proposed as a means of reducing the amount of communication required to perform consensus-based task allocation. It aims to achieve this by cutting out redundant bundle-sharing and allowing global state updates such as the configuration of DTA-related aspects of the network to be broadcast.

Additional control information is required to ensure that nodes synchronise at the end of rounds and allocation phases, to allow the clean transition from one state to another. This is provided by node selection, round completion and allocation completion messages, which inform the affected nodes of a state change requiring group synchronisation. Due to the division of the network by CF-CBBA, as described in section 4.5, global state updates are not required for DTA directly, only for the adoption of specific cluster-head nominations and cluster-configurations.

4.7 Chapter Summary

Preliminary analysis of CBBA's communication characteristics has been undertaken to better inform the areas in which communication complexity and redundant messages may be reduced. The purpose of this analysis has been to provide a means of understanding the data requirements of a DTA algorithm operating on a MANET, allowing prediction of the communication and processing costs associated with DTA and the effects of the size of the network and complexity of the task assignment problem on those costs.

Building on the problem analysis in Chapter 3, this research identifies that the division of the network into clusters may reduce the number of communication events required to achieve consensus. By utilising the multicast and broadcast capabilities of wireless

communication message redundancy may be reduced. This may decrease usage of network resources, further reducing the cost of task allocation on the network.

CF-CBBA has been proposed as a means of dividing workload between clusters instead of performing CBBA on the whole network at once. It is proposed that by performing an initial task allocation process between cluster-heads, the problem domain may be reduced in size. By reducing the number of nodes involved in this process, it is also theorised that the number of communication events required to reach consensus will be reduced.

The resulting bundles created by this initial allocation phase are then treated like task lists by the clusters belonging to the designated cluster-head. This allows further division of the problem into bundles on each node in the network while maintaining a low number of nodes involved in each parallel instance of task allocation. As a result, a global solution is computed as a result of each cluster forming a unique task list for each member node with no duplication of work.

BECF-CBBA is proposed as a means of making use of the broadcast and multicast capabilities of wireless communication to reduce communication redundancy. Unicast communication of bundles requires that nodes repeatedly communicate the same bundle in homogenous networks in which all nodes may participate in the task allocation process. This holds true for CF-CBBA, where clusters will still repeat bundle transmissions if communicating in a unicast manner. By defining state updates as global or incremental, the appropriate mode of communication may be utilised, allowing a variable level of control and redundancy reduction as it is required.

Chapter 5 will detail a test plan to evaluate the effectiveness of each proposal. The results of these tests are presented and analysed to quantitatively evaluate the effect of CF-CBBA on communication events. BECF-CBBA is analysed in terms of communication complexity (number of messages required to reach consensus), including the addition of control messages as defined in Section 4.6.

5 TESTING & RESULTS: OPTIMISED DTA

5.1 Chapter Introduction

Chapter 4 presented two novel strategies for reducing the communication overhead and complexity of CBBA. Preliminary analysis of CBBA's performance under a variable number of nodes highlighted a cubic rise in the communication events required to achieve a state of consensus. The attributes of unicast, multicast and broadcast communication were also analysed. The need for additional control communication when broadcasting was identified, highlighting the need for additional control communication.

CF-CBBA is proposed as a means of reducing complexity by limiting the number of nodes involved in any one assignment process. BECF-CBBA is proposed as a method of reducing communication complexity by allowing controlled use of multicast and broadcast communication to distribute state information throughout the network during task allocation.

Both proposed algorithms are tested, analysed and discussed in this chapter. Simulation is used to provide quantitative assessment of their performance in terms of communication events and time to convergence. CF-CBBA's task allocation optimality is also analysed to provide a basis for comparison with baseline CBBA.

The quantitative results of these investigations will provide a basis for analysis and the assessment of each algorithms attributes in terms of communication cost and potential impact on related network resources. This will allow the suitability of the proposed algorithms for use in resource constrained autonomous MANETs to be determined.

5.1.1 Chapter Layout

This chapter is laid out as follows:

- Section 5.2 discusses the experimental methodology, identifying testable elements of CF-CBBA, providing experiment plans and a defined rationale for testing and analysis.
- Section 5.3 shows experimental results for CF-CBBA compared with CBBA.
- Section 5.4 provides the experimental methodology for BECF-CBBA, compared against CF-CBBA and CBBA.
- Section 5.5 reports on test results for BECF-CBBA, providing a basis for further analysis and discussion regarding the communication event and time requirements of the proposed algorithm.
- Section 5.6 provides discussion and analysis of the results.
- Section 5.7 summarises the chapter.

5.2 Experimental Methodology: CF-CBBA

To test CF-CBBA, the rationale for testing, testable elements and resulting variables must be identified. Chapter 3 Section 3.4 identifies a series of hypotheses, of which the following applies specifically to the experiments outlined in this section:

The competing demands of control and security services may exceed the capabilities of the limited network resources available to MANETs. By reducing the complexity of communication required by DTA, the network resource requirements of control traffic for this service may be reduced, allowing greater tolerance of loss and other adverse network conditions.

The identified purpose of CF-CBBA is to reduce communication complexity. Therefore, all testing will be performed under the rationale that communication complexity must be observed under CBBA and CF-CBBA to determine if the matter algorithm fulfils this objective.

5.2.1 Test Environment and Testable Elements

To perform experiments that allow the effects of CF-CBBA on communication complexity to be observed, a simulation will be created to represent a mission area, a network of nodes and a selection of tasks scattered in the mission area. This will allow for the simulation of task allocation prior to task execution, recording the communication phase required to drive the network to consensus through bundle sharing between nodes.

MATLAB is used to provide an environment for the simulation and quantitative analysis of CBBA and CF-CBBA. A MATLAB implementation of baseline CBBA, designed by Ponda et al, is available from the MIT Aerospace control laboratory. This implementation has been modified with regards to its presentation, removing the node position in 3d space output and instead reporting on the time to solve a given problem and the number of communication events required to achieve consensus. Additional statistics tracking has been added for CBBA rounds and mission score.

A simulation environment has been created in MATLAB to provide a means of analysing the effects of the following variables on the optimality, time and communication requirements of consensus-based task allocation:

- Number of nodes.
- Number of tasks.
- Network topology.
    - Clustering.
    - Number of hops between nodes.

These attributes have been identified as the key effectors in CBBA task allocation. Equation 3-1 in Chapter 3, Section 3.2, and Equation 4-1 in Chapter 4, Section 4.5, highlight the importance of the number of nodes in determining the number of communication events required to achieve consensus. In that same section, the effect of a reduced task list is also discussed, highlighting the potential benefits of problem sub-division between clusters when considering the number of bytes sent. As a result, the number of nodes and number of tasks are both considered critical variables when analysing the communication cost, assignment optimality and network resource

utilisation of a given network performing a CBBA-like process. As such, they have been selected as the primary variables for the investigation.

The basic configuration of the simulation is as follows:

Table 5-1 Table showing the basic configuration of the MATLAB simulation for CF-CBBA

Attribute	Value
Mission Area	100m x 100m
Number of Tasks	1-50
Number of Nodes	18
Task Score	100
Node communication range	50m
Maximum hop count	5
Number of iterations	100
Seed	11

A mission area of 100 by 100 meters has been selected to provide a clear boundary to the area in which nodes and tasks are spawned at the beginning of the simulation. This provides a space in which tasks and nodes are placed randomly. The size of the chosen space provides a wide variety of different sets of initial conditions across the multiple iterations of the simulation. This same mission area is used by Brunet and Choi in their respective publications (Brunet et al. 2008) and (Choi et al. 2009).

A range of 1 to 50 tasks provides a sample over which the characteristics of the network can be observed. A limit of 50 tasks has been set so as to not over saturate the mission area with tasks. This limit has also been chosen to provide a set of tasks which does not by sheer number of tasks, force nodes to have to become oversubscribed (that is possess more than five tasks in a bundle in the final allocation). Brunet, Choi and Ponda use task counts up to 40 in their simulations and experiments. A slightly larger problem has been selected here to allow the observation of reduced optimality in larger networks, as the 40 task problem domains did not offer as much in the way of comparative analysis of different network sizes during preliminary modelling of CBBA across different network sizes.

The total number of nodes has been set to 18 for all experiments. This is number has been chosen based on preliminary experiments using varying numbers of nodes to determine

the minimum effective set of nodes required to complete task allocation without guarantees of over-allocation. The minimum identified set of nodes, outlined in Section 4.3, was 6 nodes for 50 task allocation, before optimality dropped below 45%.

CF-CBBA is compared with CBBA using networks of 18 nodes, to maintain comparable network sizes between the two DTA approaches to ensure the validity of observations made as a result of the simulations performed on both algorithms. This provides a sample that adheres to the guidelines identified above, and that is divisible into clusters of 3 or 6 nodes. This will allow the effects of different sizes and numbers of clusters to be compared, as well as allowing comparison of CF-CBBA and CBBA. A variable number of tasks between 1 and 50 are used to determine the effects of increasing problem complexity on the network.

Additionally, there are minimum-distance considerations to take into account if attempting to model real world scenarios. The CAP 722 outlines the accuracy of GPS modules, determining that a minimum 7.8 meter distance is required to reliably avoid collision when using GPS-only collision avoidance techniques in micro-UAVs. In networks with 20 or more nodes, the distribution of nodes throughout the mission area without a high probability of two nodes being placed within this safe-zone. As a result, 18 nodes was selected as the sample size to allow multiple clusters of nodes to be reliably formed, while avoiding over-crowding of the mission area.

To maintain parity with existing publications regarding CBBA, task scores are set to 100. This score is subject to the costs associated with traveling between tasks and DMG as previously outlined in Chapter 4, Section 4.3.

To analyse the effects of multi-hop communication, a 50 metre communication range has been allocated to all nodes. This will leave the network largely connected, but with some multi-hops routes to provide variability between iterations and demonstrate the additional communication cost of retransmission. This range has been chosen to limit the length of routes and ensure that scenarios in which nodes are disconnected from the network entirely at initialisation are very unlikely. A maximum hop count of 5 has been chosen to ensure that nodes do not relay messages more than four times. Due to the combination of mission area and communication range, it is unlikely that hop counts will exceed a value

of 2, but this cap has been implemented to prevent continuous looping of communication, which is possible when no bounding value is added to the routing process.

In the case of such networks being generated, the network is regenerated to ensure communication is possible across the whole network. It is considered vital that nodes are connected to the network by at least one other neighbouring node to allow their inclusion in the DTA process.

All experiments are run 100 times, from 1 to 50 tasks. The average of all iterations is taken to provide data points for all output generated during these experiments. Data points are generated for all 50 task values that are used to provide a point of reference for comparative analysis. The common seed value is used as the basis for generating the vector of random values, which will be used for all experiments to provide a consistent set of scenarios between all networks. This ensures that task and node placement remains consistent in all scenarios.

These attributes remain constant throughout the testing process. This also provides a means of describing network topology, as the range characteristic of each node provides an indication of connectivity. To calculate the graph of the network, Dijkstra's shortest path algorithm is used. All experiments are performed on fully networked groups of nodes. Any node that can connect to all others via routes no longer than the hop count of five are considered to be networked.

As a cluster-based extension of CBBA, CF-CBBA requires not only comparative analysis against baseline CBBA, but analysis of the effects of varying cluster sizes. This is required as the optimality of CBBA task allocation has been shown to decrease with a larger node to task ratio. It is therefore important that the effect of the number of cluster-heads and member nodes involved in an assignment is analysed.

Two cluster-sizes have been selected to allow for a comparative analysis of the effects of different cluster arrangements on network and DTA characteristics (such as communication events, bytes transmitted and assignment optimality):

- Three clusters of six agents.
- Six clusters of three agents.

These clusters have been selected to allow the sub-division of the 18 node network into even clusters. By choosing two different cluster distributions, the effects of clustering can be analysed in addition to the comparison of CF-CBBA with CBBA. This is useful, as it is possible that the division of the network into clusters may affect communication events, time to complete and the optimality of the assignment.

In each case the number of cluster-heads is always equal to the number of clusters. Cluster-head selection is performed a priori, the method of selection is not analysed as it has no bearing on the role of the cluster-head in task allocation. Although such methods are out of the scope of this research, it is possible to designate cluster heads based on node energy levels, location relativity to other members of the network, or other differentiating features such as higher computational capability or node connectivity.

It is also possible that CBBA may be used to treat nodes as resources to be allocated, using the previously mentioned attributes to score nodes to rate their fitness to fulfil the user-defined requirements of a cluster head. So long as cluster heads are able to effectively complete an initial round of task allocation, and pass it down to their cluster in a timely manner, they meet the minimum requirements of this role.

For the purpose of simulation, the common centre of the cluster is used to place bids. This is the average of all X and Y values in the cluster. As previously identified in Section 4.5, this is done to reduce the effects of node distribution throughout the mission area on the suitability of tasks. By using the centre of the cluster as a common variable for bids, the cluster-head avoids biasing the task allocation process towards itself or specific nodes in the cluster. Outlier nodes may still be disadvantaged by their position.

Each experiment is performed 100 times, over 1-50 tasks to ensure a wide-range of potential node and task placements are observed within the simulation-space. This provides a large sample of different network configurations, allowing general trends based on the variables to be observed.

5.2.2 Expected Output for Analysis

Results relating to communication complexity can be defined as the number of communication events required to achieve consensus, the network resources required to facilitate the communication of the DTA process, and the optimality of the resulting assignment.

Communication events are recorded every time a network exchange or global control message is required, incrementing a global communication event counter with every such instance. This provides a running total of communication events throughout each experiment, cleared for every increment of task set size to ensure that values are representative. All values for communication events are shown as integers, rounded up to the nearest whole number after averaging all iterations. This reflects the need for complete communication events; it is not possible to have a partial communication event.

Network resource utilisation is measured by tracking the number of bytes sent when communicating bundles. The number of bytes sent gives a direct measurement of the network resources required by the DTA process. This is affected by the number of nodes and the number of tasks in a given problem domain. As the size of the problem domain affects the total number of bytes sent per communication event, the effects of cluster heads dividing the problem domain prior to passing their bundles down to their clusters as a new problem may reduce the number of bytes required by each communication event.

Each task is broken down into three integer values when communicated; task ID, current winning bid ID, and current winning bid value. The X, Y and Z coordinates of each tasks are held locally on a task list provided at initialisation to each node, so do not require communication. Each task is therefore 12 bytes in length when sent in its short form as described above. The total payload length of a packet will therefore be 12 bytes multiplied by the number of tasks. Each packet includes a 32 byte 802.11 header and an 8 byte UDP header for each communication event. For example, a 50 task problem is 640 bytes in length, resulting in all packets falling within the standard Maximum Transmission Unit (MTU) of 1500 bytes common to most network interfaces. This means that multiple packets are not required to transmit any bundle in the network.

Mission optimality is computed using the built in mission score feature of the CBBA MATLAB implementation. Each task is assigned a base score of 100, with the cost of moving to that task (1 point per metre) being subtracted from the score to calculate each bid placed on a task by each node. The winning node, assuming it completes the task, scores points equal to the value of its bids. DMG is applied in all cases, meaning that tasks lower in a bundle will be reduced in value to enforce the ordering of tasks in each bundle. The sum of all node scores represents the network score for the mission, which is used to determine the optimality of the assignment. Higher values are more optimal, as they come closer to the initial value of all tasks. The highest possible score is 5,000 for 50 task problems, as each task is worth 100 points in perfect conditions (no cost to travel, no DMG).

The hardware used to conduct these experiments has the following specifications:

- Intel i7 3.4GHz processor.
- 64 GB DDR3 1333MHz RAM.
- 250 GB Solid State Hard Drive.

Software specifications are:

- Windows 7 64-bit Operating System.
- MATLAB R2013a.

These specifications remain constant throughout all experiments to ensure that no underlying characteristics affect the outcome of one simulation differently to another.

Experimental results are provided for CF-CBBA and BECF-CBBA. Both algorithms are analysed to identify their performance characteristics and these are compared against those of CBBA in the following sections.

5.3 Results: CF-CBBA

The following results compare CBBA against two different configurations of CF-CBBA. The compared algorithms are, as outlined in Section 5.2:

- CBBA.
- CF-CBBA: three clusters of six nodes.
- CF-CBBA: six clusters of three nodes.

The comparison of CBBA to CF-CBBA requires that CF-CBBA divides the network into clusters, to break down the number of agents involved in any given instance of task allocation. The intention of this approach is to reduce the time taken to achieve consensus and reduce the total number of communication events required to facilitate the DTA process.

Two cluster formations have been selected to allow each configuration to be compared against CBBA, and provide data for a comparative analysis of different network configurations using CF-CBBA. This will, in turn, provide information on how the CF-CBBA algorithm responds to the size of the problem domain in different configurations. In all cases, a network size of 18 nodes is maintained.

In all cases the number of cluster-heads is equal to the number of clusters. Results are shown for the following:

- The total number of communication events required to achieve consensus.
- The total number of bytes transmitted during the DTA communication phase.
- The total mission score of the network.

All clusters are formed prior to task allocation in these experiments.

5.3.1 Number of communication events to reach consensus

The number of communication events required to achieve consensus is shown in Figure 5-1. Baseline CBBA requires 306 events for 18 nodes attempting to assign a single task amongst themselves. This rises to 1224 events for 50 tasks.

CF-CBBA in a three cluster configuration requires a total of 36 events to allocate one task, reaching a peak of 216 events for 32 tasks. 202 events are required for 50 tasks. The initial drop from 306 events for one task under CBBA, to 36 events in this configuration

of CF-CBBA appears to be abnormally large, but the underlying way in which communication is managed by CF-CBBA provides an explanation.

Under CBBA, all nodes are required to participate in the single task allocation, even if they have no chance of actually receiving it. However, under CF-CBBA, only cluster heads are forced to participate in this manner, with the winning cluster head then performing another round of allocation within its cluster. Any nodes in clusters that did not receive any tasks do not perform CBBA, and therefore do not contribute to the total communication event count.

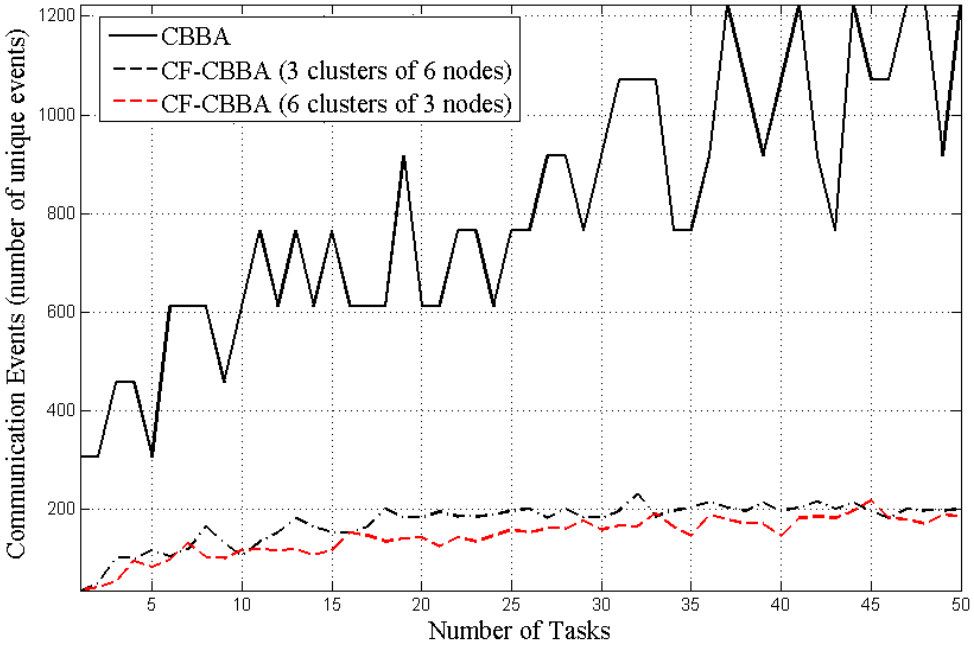

Figure 5-1 Graph comparing the number of communication events required by CBBA and two cluster-formations of CF-CBBA over an increasing number of tasks

This is broadly comparable to the six cluster configuration, which requires 36 events initially, reaching a high of 196 events for 50 tasks. The six cluster configuration follows a trend of requiring less communication than the three cluster configuration, but with broadly similar characteristics. Both configurations of CF-CBBA require substantially less communication than CBBA.

5.3.2 Network resource utilisation

Figure 5-2 shows the number of bytes transmitted by CBBA and CF-CBBA over the course of task allocation. Two key variables affect the total number of bytes sent; the number of communication events and the length of bundles transmitted. Bundles are a list of tasks the length of all tasks available for bidding, minus those that have a higher bid submitted by another drone. This means that as the network reaches convergence, the number of tasks in transmitted bundles will decrease, as nodes will stop being able to outbid for tasks that have been claimed by the globally optimal node for that task. Furthermore, in clustered scenarios, the problem domain is reduced is size for the cluster phase of allocation due to the initial division of labour between cluster heads.

CBBA performed on 18 unclustered nodes requires 88 kilobytes of data to be sent to achieve consensus for a single task. This is largely due to the size of the 802.11 packet, as the single task is only 12 bytes long. A brief spike in data transmitted is seen at 4 tasks, with a rise to 26 kilobytes. This is caused by a larger number of CBBA rounds being required to achieve consensus. A maximum data send of 2.1 megabytes is reached for 50 tasks, by CBBA.

CF-CBBA in a 3 cluster configuration with 6 nodes per cluster requires 3.6 kilobytes to converge to a solution for 1 task. This is a significant drop, caused by the smaller number of nodes contributing communication events at each phase of allocation, and by the fact that only one cluster will perform cluster-level CBBA, resulting in a much lower amount of data being sent in total. When assigning 50 tasks, CF-CBBA in this configuration requires 56 kilobytes to achieve consensus, markedly lower than CBBA.

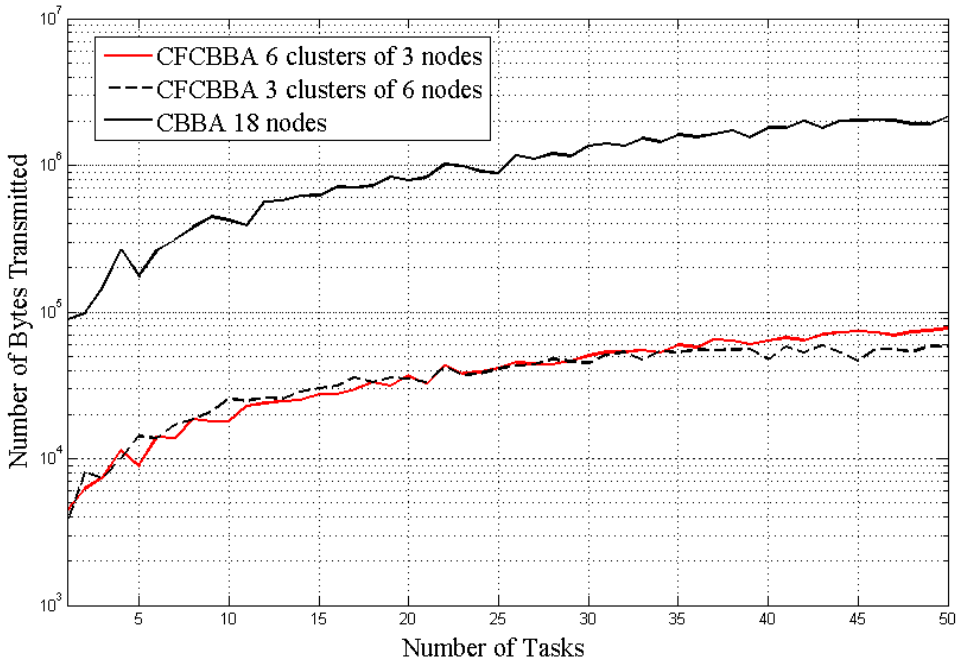

Figure 5-2 Number of bytes transmitted during DTA by CBBA, and CF-CBBA in two cluster configurations

CF-CBBA has also been tested in a 6 cluster configuration with 3 nodes per cluster. Notably, the results for this configuration closely match those for the 3 cluster configuration. Initially, this configuration requires more data to be transmitted, needing 4.3 kilobytes to achieve consensus for a single task. This is due to the larger cluster head size, 6 nodes participating in the initial allocation phase means a higher number of communication events. Between 4 and 28 tasks, 6 cluster CF-CBBA outperforms 3 cluster CF-CBBA marginally, before rising to 77.3 kilobytes to allocate 50 tasks.

This configuration is still a significant improvement over CBBA, with comparable data requirements to 3 cluster CF-CBBA for the majority of problem domain sizes. Larger problem domains begin to show the additional data requirement of 6 cluster CF-CBBA driven by the more complex cluster head allocation stage when compared with 3 cluster CF-CBBA. Both configurations send notably less data than CBBA.

5.3.3 Optimality of the resulting assignment

Figure 5-3 shows the effects of clustering on the optimality of an allocation. CBBA sets the baseline trend by representing a completely unclustered network of 18 nodes. Initially scoring 195 points for 2 tasks, CBBA achieves 97.5% optimality. At 50 tasks, it achieves a score of 4812.25 points, which is 96.2% optimal. Minor deviation from a linear trend is visible, but the optimality of assignments does not significantly degrade over the course of this experiment.

CF-CBBA in a six cluster configuration shows signs of assignment degradation after 30 tasks. An optimality of 97.5% is achieved with 2 tasks and this is maintained up to 30 tasks. At this point, the optimality of the assignment becomes variable, with a low of 86.2% at 40 tasks and an optimality of 92.3% at 50 tasks.

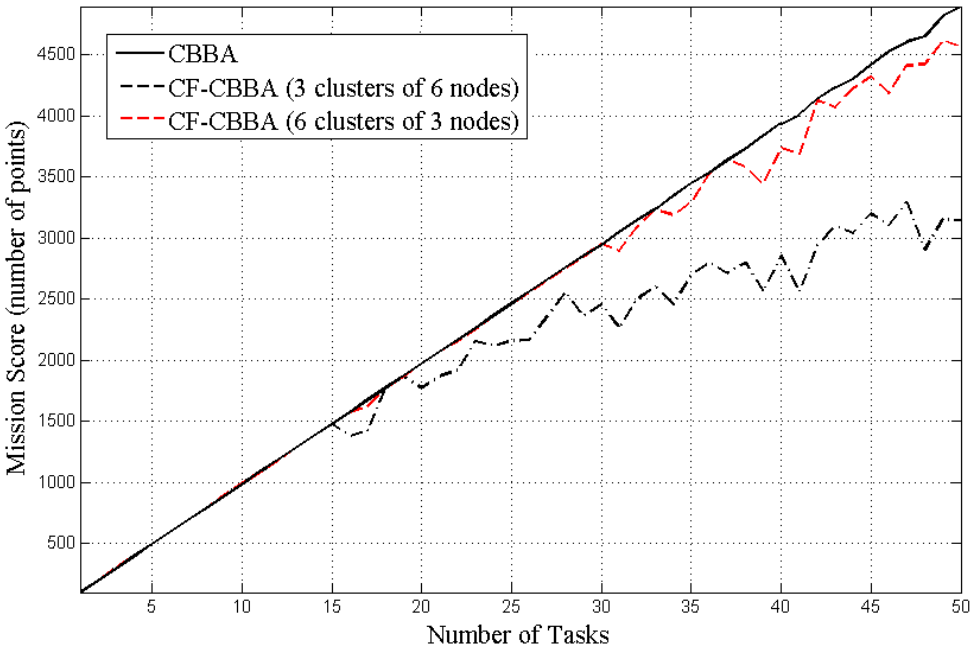

Figure 5-3 Graph comparing the optimality of task allocation under CBBA and two cluster-formations of CF-CBBA over an increasing number of tasks

The three cluster configuration loses optimality after 15 tasks. As with the other two configurations, the three cluster test initially achieved an optimality of 97.5% for 2 tasks. At 17 tasks this drops to 84.4%, beginning the variable trend shared by the six cluster configuration. At 50 tasks, the three cluster configuration is only 62% optimal and appears to be in rapid decline in terms of its ability to service the task set.

5.4 Experimental Methodology: BECF-CBBA

To test BECF-CBBA, the rationale for testing, testable elements and resulting variables must be identified. Chapter 3 Section 3.4 identifies a series of hypotheses, of which the following applies specifically to the experiments outlined in this section:

DTA algorithms require a large amount of communication to reach a global solution. By using consensus-based methods and adapting them to make use of the unique communication and topological properties of MANETs, it will be possible to reduce the communication overhead of DTA in autonomous MANETs.

The rationale for testing is therefore focused on identifying the effects of broadcast communication on communication complexity. This requires a similar set of testable elements to those discussed in Section 5.2.

5.4.1 Test Environment and Testable Elements

BECF-CBBA is compared against CF-CBBA and CBBA in networks of 18 nodes, to maintain continuity and allow for comparison with the experiments conducted in Section 5.3. The smaller number of nodes is used to allow observation of trends where the number of tasks is low.

Section 5.3 shows that networks configured in a 6 cluster, 3 nodes per cluster, manner provided the best return in terms of optimality, network resource utilisation and communication events. As a result, this cluster formation has been selected to represent the network configuration for CF-CBBA and BECF-CBBA in these simulations. BECF-

CBBA will also be tested in an unclustered configuration to allow direct comparison with CBBA, effectively providing a broadcast CBBA implementation with no network configuration changes.

CF-CBBA provides a foundation for analysis of the effects of cluster size of assignment optimality, but the analysis of BECF-CBBA focuses on communication. The optimality of an assignment is driven by the number of nodes and tasks in a given task allocation process under CBBA. As a result, the manner in which bundles are communicated has no effect on the optimality of the assignment, though it may have an effect on the number of communication events to converge to a solution.

5.4.2 Expected Output for Analysis

The outcomes to be analysed as a result of simulation can be collectively referred to as measurements of communication complexity (or the cost of communication). Communication complexity can be broken down into communication events and network resource utilisation. Mission optimality is not a focus for these experiments as preliminary research showed that differing communication strategies (unicast, multicast and broadcast) had no effect on optimality, which was a result of the size of the network compared to the size of the problem domain at each stage of task allocation.

The number of communication events observed under BECF-CBBA is comprised of two types of communication: bundle exchange and message relaying. The latter type of communication represents the routing of packets between nodes that are out of local communication range and thus require intermediate nodes to relay communication. Relaying messages is a network service, and all network nodes may relay messages on behalf of any cluster.

As defined in Section 5.2, network resource utilisation is measured as the number of bytes required to achieve consensus during the communication phase of CBBA or an equivalent algorithm. BECF-CBBA will be compared against CBBA and CF-CBBA in unclustered and clustered networks to analyse the effects of multicast and broadcast communication on the total network utilisation under BECF-CBBA.

It should be noted that routing packets are not considered in the context of this investigation, as it is assumed that routes have been generated prior to the task allocation process. Routing cost, outside of the relaying of data packets during task allocation, is considered to be outside of the scope of this research as such costs are highly dependent on the routing algorithm in use.

5.5 Results: BECF-CBBA

The results of simulation, as defined in Section 5.4, are presented and discussed in this section. A more thorough analysis of the results is undertaken in Sub-section 5.6.2.

Results are shown for the following:

- Number of communication events required to achieve consensus.
- Number of bytes transmitted during the communication phase of the DTA algorithm being simulated.

Mission optimality is not affected by the communication being unicast, multicast or broadcast and so has not been analysed here.

5.5.1 Number of communication events to reach consensus

CBBA requires 306 communication events to allocate 1 task in this experiment. Figure 5-4 shows a peak of 1224 events is seen at 39 tasks, with 918 events required to allocate 50 tasks. As previously observed, the number of events varies greatly depending on the number of rounds required to reach a convergent state.

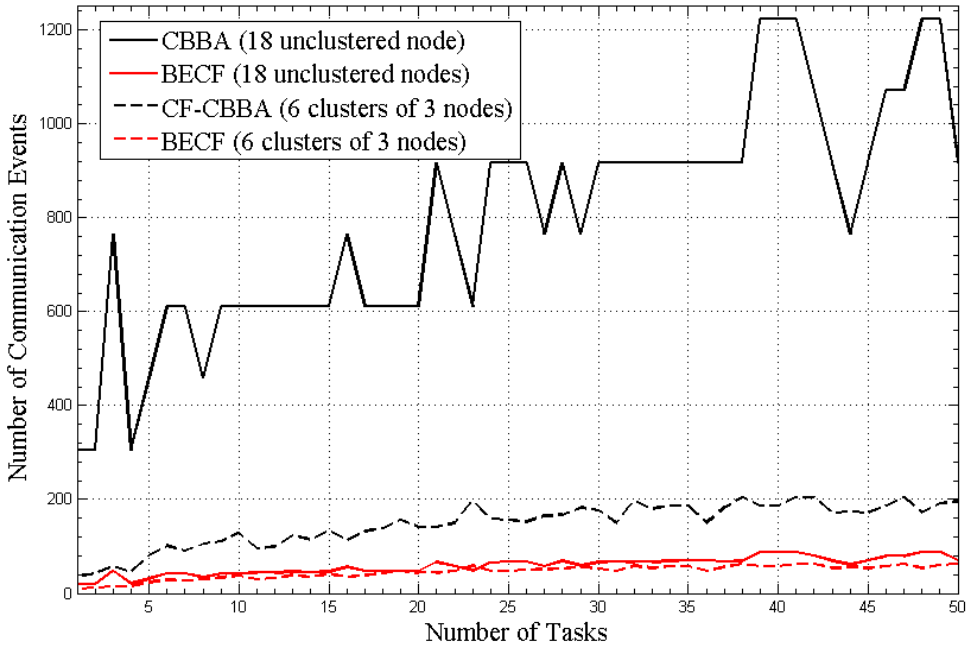

Figure 5-4 Graph comparing the number of communication events required to achieve consensus in an 18 node network using CBBA, CF-CBBA and BECF-CBBA

The influence of CBBA rounds on the number of communication events required is most notable for 3, 6, 26 and 44 tasks, all of which fall substantially below the average communication event count relative to their neighbouring task number values. As reported in section 5.3, this phenomenon is a result of the nodes in the network performing multiple rounds of CBBA to resolve conflicts between nodes, where many nodes have a similar bid for a task or set of tasks.

CF-CBBA requires 36 communication events to converge to a solution for 1 tasks in a 6 cluster, 3 node per cluster, configuration. This rises to a high of 216 events for 50 tasks. The effects of CBBA rounds is also evident here, though the smaller number of nodes involved in each phase of allocation reduces the visibility of such irregularities.

BECF-CBBA uses multicast communication to address all members of a cluster. BECF-CBBA is used in two configurations; to operate in 6 clusters of 3 nodes, identical to the CF-CBBA configuration, and an 18 node unclustered version used to allow a direct comparison with CBBA.

Including relayed messages to ensure that nodes receive multicast messages in cases where they may be out of range of the source node, 12 events are required to achieve a solution for 1 task in clustered BECF-CBBA. The effects of CBBA rounds can be seen as low level noise in the plotted line for BECF-CBBA, but at a much reduced impact when compared to CF-CBBA and CBBA. A high of 62 events is reached for 50 tasks.

Unclustered BECF-CBBA takes a similar amount of communication events, when compared to clustered BECF-CBBA, to achieve consensus. 23 events are required to achieve consensus for 1 task. A high of 121 events is seen at 49 tasks, dropping to 70 events for 50 tasks. This demonstrates that the use of broadcast communication (or multicast when using clusters), can greatly reduce the communication requirements of CBBA and CF-CBBA by reducing sequential communication. This in turn results in fewer network resources being consumed by the DTA operation, potentially making the network more tolerant of packet loss (and subsequent retries when sending bundles).

These results demonstrate that BECF-CBBA requires substantially fewer communication events than CBBA and CF-CBBA, due to reducing the amount of redundant communication required to transmit bundles to each other node in a cluster. This includes the additional communication required to relay bundles over intermediate nodes (routing data between non-local node-pairs), showing that even with additional overheads, BECF-CBBA out performs CBBA and CF-CBBA in this context.

5.5.2 Network resource utilisation required to reach consensus

Figure 5-5 shows the total number of bytes transmitted during task allocation by CBBA, CF-CBBA and BECF-CBBA. Two network configurations have been chosen, as outlined in Section 5.4; an 18 node unclustered network and a 6 cluster 3 node per cluster configuration. The latter configuration has been chosen as a result of observations made in Sub-section 5.3.1.3, where the optimality on configurations with 3 cluster heads was highlighted as being far below than of 6 cluster networks.

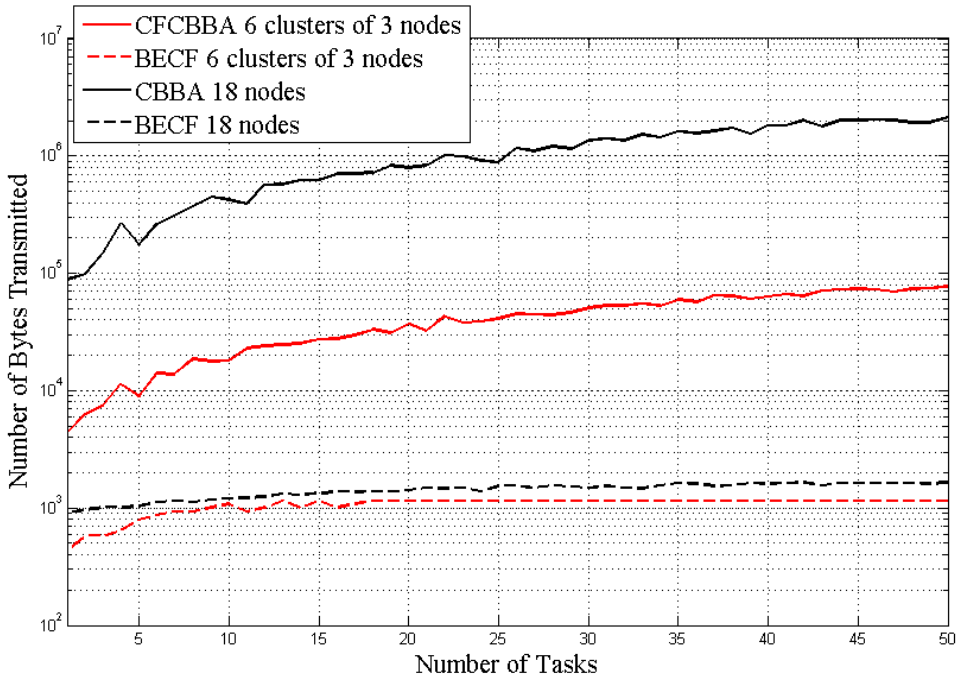

Figure 5-5 Number of bytes required to achieve consensus, under CBBA, CF-CBBA and BECF-CBBA

CBBA and CF-CBBA perform as they did in Sub-section 5.3.1.2, with CF-CBBA sending much less data than CBBA for the same number of tasks throughout all problem domain sizes. This demonstrates the effects of differing network configurations on the DTA process, but nodes not demonstrate the effects of broadcast and multicast communication on the amount of data sent.

BECF-CBBA, in both clustered and unclustered networks, significantly reduces the amount of data sent by removing redundant communication. In unicast networks, CBBA must repeatedly contact nodes and send the same information to each node, to allow the network to cooperate and move towards a convergent state. BECF-CBBA allows a single transmission to be sent to all target nodes (multicast) or to the entire network (broadcast) with that information, removing the need to send the same information repeatedly in a serial manner.

Unclustered BECF-CBBA requires only 912 bytes be sent for a 1 task allocation problem, with a maximum of 1.6 kilobytes for 50 task problems. Clustered BECF-CBBA, in a 6 cluster configuration with 3 nodes per cluster, requires 432 bytes to converge on a

solution. A maximum value of 1.2 kilobytes is observed for 50 tasks, with a flat trend from 18 tasks. This flat trend is caused by the distribution of tasks among nodes reducing the size of bundles rapidly. The smaller problem domains passed down to the cluster level result in a rapid decrease in the size of the bundles transmitted by each node during the cluster allocation phase of BECF-CBBA and CBBA. This results in smaller packets, the required number of which is decreased to a minimum by the use of broadcast communication

5.6 Discussion

The characteristics of CF-CBBA and BECF-CBBA will be discussed relative to the hypotheses that led to their proposal and preliminary analysis undertaken in Sections 4.3 and 4.4. The effectiveness of these algorithms will be analysed in terms of their ability to deliver optimal solutions to task allocation problems with a minimum of communication events.

5.6.1 Analysis of CF-CBBA

5.6.1.1 Number of Communication Events

Figure 5-6 provides a comparison of CBBA with the two cluster configurations of CF-CBBA, showing the percentage of CBBA events required to achieve consensus under each configuration, when compared against CBBA networks comprised of 18 nodes. This allows the identification of the relative number of communication events required to drive the network to consensus. By identifying this attribute of each CF-CBBA cluster formation, the effects of clustering on the network when performing DTA can be observed and analysed.

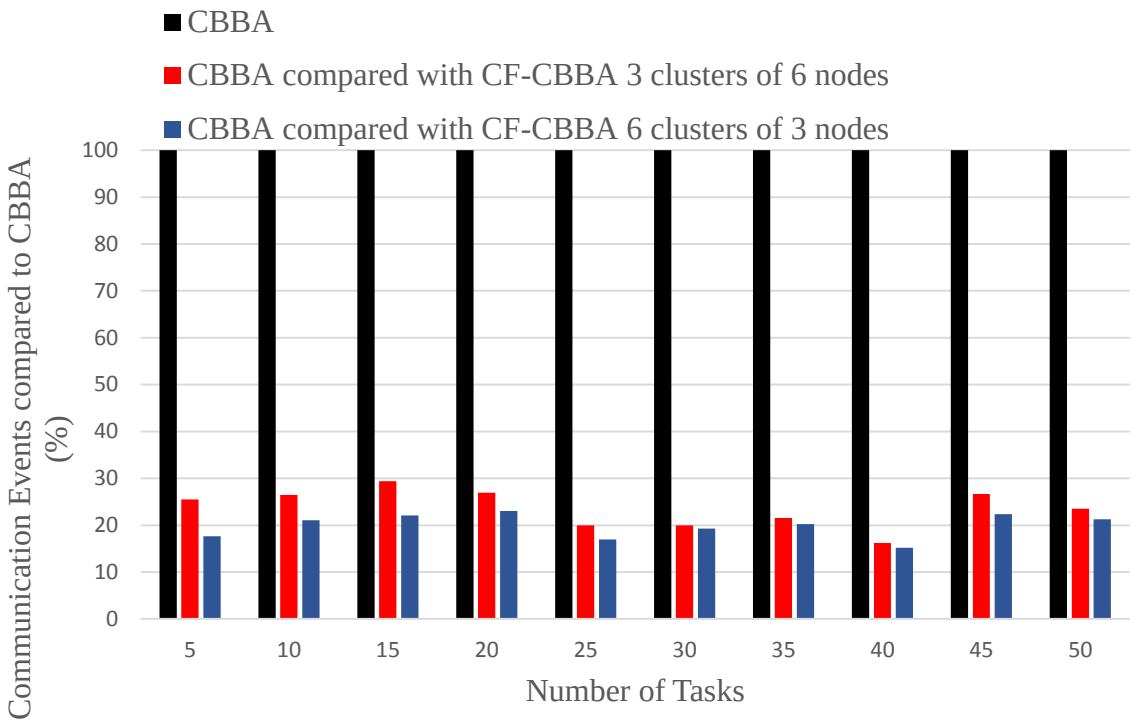

Figure 5-6 Chart comparing the difference in communication events between CBBA and CF-CBBA

In all cases, CF-CBBA requires significantly less communication events to achieve consensus than CBBA. There are, however, notable findings when comparing the trends of the two CF-CBBA configurations.

Both configurations show seemingly anomalous decreases in communication events at 25 and 40 tasks. It has been observed previously that the number of rounds required by CBBA varies with the number of nodes and tasks. In this case, fewer rounds are required as allocation is able to be completed within a very small number of rounds, due to the ease of division of the problem domain between all members of the network.

The division of tasks between nodes can also lead to tie-breaker situations, wherein two or more nodes place similar bid values on a task. This can be observed for networks of 3 clusters comprised of 6 nodes at 15 tasks, and for networks of 6 clusters of 3 nodes at 20 tasks.

These anomalies are driven by contention and the need for additional rounds in which the tie is broken by comparing the value of existing tasks in the competing node bundles to

determine the highest potential score achievable by the allocation of the task to one of those nodes. This leads to additional communication to facilitate this sorting process, as seen in Figure 5-7.

Figure 5-7 compares the two configurations of CF-CBBA more closely, focusing on the difference between the communication events required by each network.

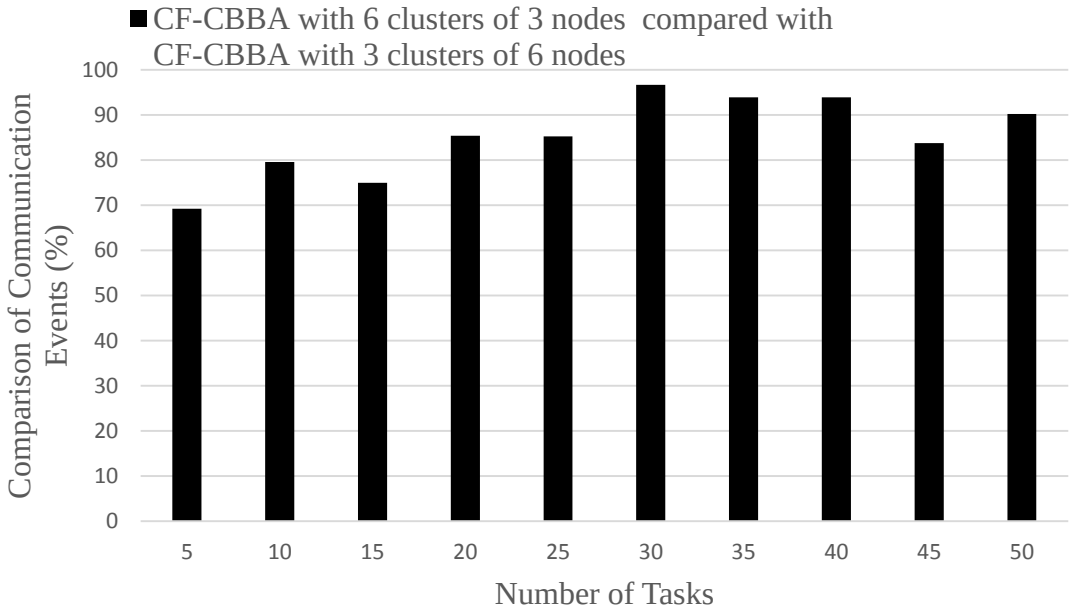

Figure 5-7 Chart showing the number of communication events required by CF-CBBA using 6 clusters of 3 nodes compared against a 3 clusters of 6 nodes network

The effects of cluster configuration on the number of CBBA rounds required to converge is seen here for 15 tasks. The difference is pronounced here, with 6 cluster CF-CBBA requiring 75% of the communication events required by CF-CBBA 3. The largest difference is seen in the allocation of a single task, with 6 cluster CF-CBBA requiring only 69% of the communication needed by 3 cluster CF-CBBA.

Due to the higher number of cluster-heads in the 6 clusters of 3 nodes configuration, the initial communication requirement of that configuration is generally higher than that of the 3 clusters of 6 nodes network. This is driven primarily by the need to process the entire task list at the cluster-head stage; the 6 clusters of 3 nodes network will break the problem down into smaller lists for the cluster allocation process, but must first process the entire problem among 6 cluster heads. However, this cluster head level complexity

does not appear to propagate to the cluster level allocation, as shown by the lower communication event counts shown above.

The trend observed in Figure 5-7 shows that 6 clusters of 3 nodes networks require less communication than 3 clusters. Although communication is more complex at the cluster-head level for 6 clusters, the smaller problems on each cluster when performing inner-loop allocation, and the smaller cluster sizes, reduce cluster level communication significantly. As the number of nodes is more significant than the number of tasks when determining the potential communication requirements of CBBA, this has a pronounced effect, especially when considering allocations in which many rounds may be required.

The observed difference in communication events between CF-CBBA on networks comprising 6 clusters of 3 nodes and 3 clusters of 6 nodes is generally not large, due to the need for both networks to compute a solution at both the cluster-head and member level. The higher cost of communication at the cluster-head level for 6 clusters of 3 nodes networks is comparable to the more complex cluster level communication of 3 clusters of 6 nodes networks, leading to generally similar communication requirements in large problem domains. It is only when the number of rounds is positively affected by the combination of nodes and tasks that gains of over 15% are observed in CF-CBBA network with 6 clusters of 3 nodes. Four such cases can be observed in problem domains comprised of 5, 10, 15, and 45 tasks.

5.6.1.2 Number of Bytes Transmitted

Figure 5-8 compares the number of bytes transmitted by CF-CBBA in a 6 cluster, 3 node configuration and a 3 cluster, 6 node configuration. This graph allows for a close inspection of the respective data requirements of both algorithms under unicast communication, allowing for a more in-depth analysis of the different network configurations than the results shown in Sub-section 5.3.1.2 provide.

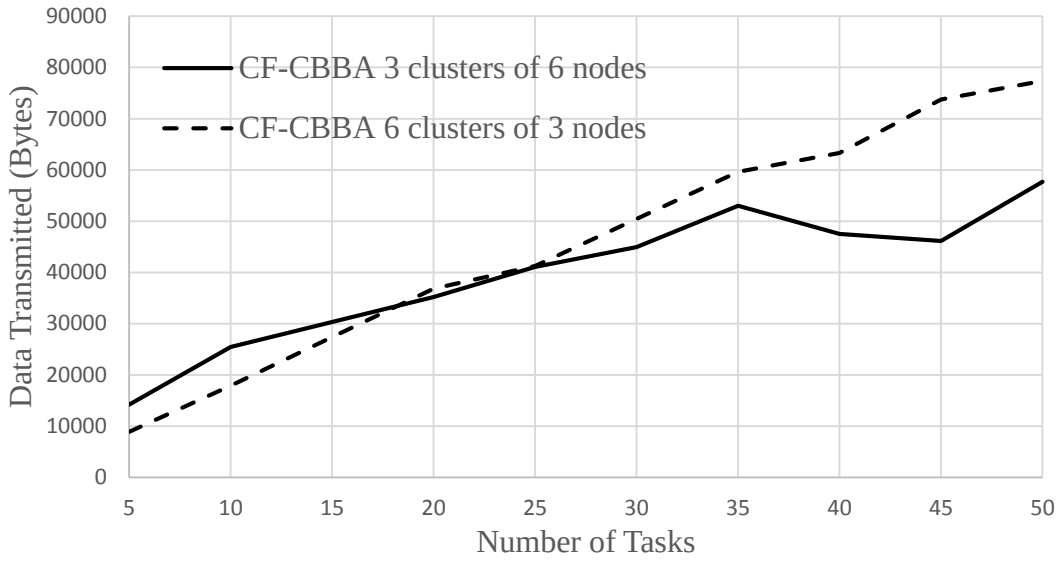

Figure 5-8 Graph showing the number of bytes transmitted while performing CF-CBBA on network of 3 cluster of 6 nodes and 6 clusters of 3 nodes

The difference between 6 clusters of 3 nodes and 3 clusters of 6 can be clearly identified at 5 tasks. 6 clusters of 3 nodes require less than 10 kilobytes of data to converge for this number of tasks, compared to the 13,512 bytes required by 3 clusters of 6 nodes. This is driven by the involvement of clusters, at this stage. In the 6 cluster network, not all nodes will be involved in the cluster level allocation stage, due to there being only 5 tasks to go around. However, the 3 cluster network is likely to involve all clusters, as there will be a high likelihood of each cluster qualifying for at least one task to be passed to the cluster level.

This trend of 6 cluster networks requiring less data drops off at 18 tasks, the point at which all clusters in the 6 clusters of 3 node networks begin to receive tasks t process at the lower level. It is at this point the more complex communication requirements of the cluster head allocation phase become apparent for such networks.

CF-CBBA performed on 3 clusters of 6 nodes begins to require less data to converge to a solution after 25 tasks. At this stage the simpler cluster-head allocation phase starts to become apparent, as problems are sub-divided with a minimum of communication events due to the low number of nodes involved. At 50 tasks, 3 clusters of 6 node networks require only 56 kilobytes compared to the 77.3 kilobytes required by networks comprised of 6 clusters of 3 nodes.

5.6.1.3 Optimality of the Assignment

Figure 5-9 visualises the optimality of task assignment under CBBA, and two configurations of CF-CBBA. CBBA follows a general trend of delivering solutions of approximately 98% optimality, accounting for the cost of travel to each task and the effects of DMG on score as bundles become large on individual nodes.

CF-CBBA, in a network made of 6 clusters of 3 nodes, maintains an approximately 98% assignment optimality until 30 tasks. A low of 91.28% is reached at 50 tasks, after a steady decline in optimality from 98.3% at 30 tasks. An anomalous result is seen at 45 tasks, with the optimality of assignment rising to 96%. This is due to the distribution of tasks among nodes having more often than not resulted in as even a distribution as possible, minimising the effects of DMG on the assignment while maximising the local optimality of each task assignment.

CF-CBBA, in a network made of 6 clusters of 3 nodes, rapidly declines in optimality after 15 tasks. A drop from 98.3% to 62.9% is observed between 15 and 50 tasks. This drop is caused by the over-allocation of tasks to individual nodes at the cluster-head level. In Section 3.2, the optimality of assignment was shown to be directly linked with the number of nodes involved, leading to a conflict between optimality and cost of communication. This is visible here, as the 3 node cluster-head allocation can result in one node being allocated significantly more tasks than the other two due to an advantageous position from the point of view of bidding. The resulting allocation is then passed down to the cluster level, where it may again result in over-allocation to specific nodes.

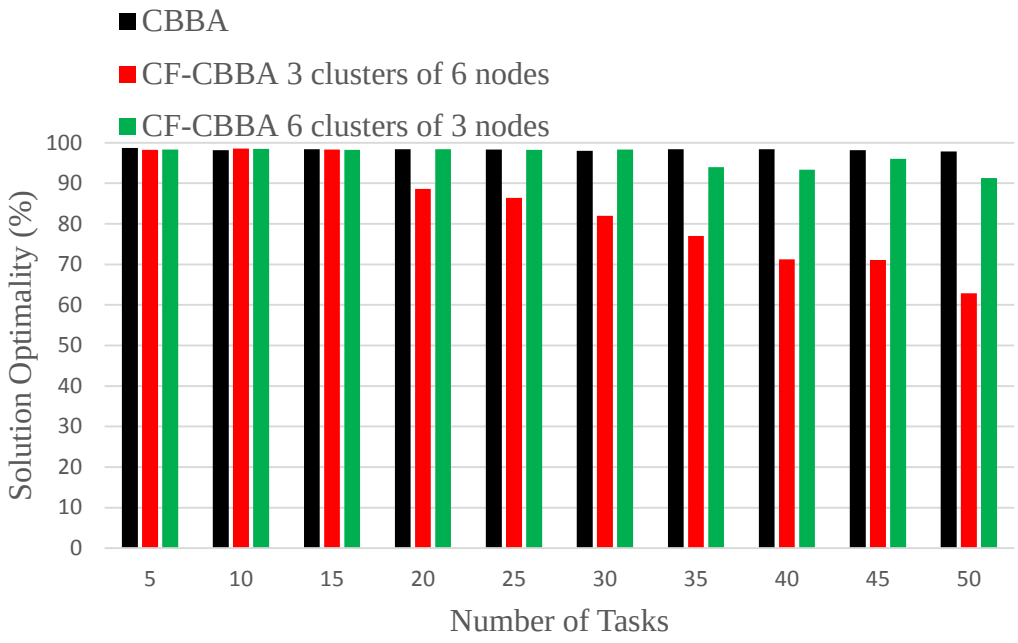

Figure 5-9 Chart comparing the optimality of solutions derived using CBBA, 3 cluster and 6 cluster CF-CBBA

5.6.1.4 Section Summary

CF-CBBA has been compared with CBBA and analysed to determine its characteristics in terms of required communication events, and assignment optimality.

CF-CBBA has been found to require less communication to achieve consensus. This is due to the reduction of the number of nodes involved in each stage of allocation. Despite the need to run CBBA twice, once in an outer loop for cluster-head allocation and again in an inner loop for cluster member allocation.

The grouping of nodes into clusters was shown to have an effect on the number of events with a larger number of cluster-heads reducing the total number of events. This has been analysed and found to stem from the increased communication complexity in networks with a larger number of nodes, as routing between cluster-heads is vital to ensure reliable communication of bundles.

Similarly, the division of the network into clusters was shown to have an impact on the data transmission required to achieve a state of consensus. Two key contributors to the

reduction of transmitted data have been identified; the complexity of communication at the cluster head and cluster level of allocation, and the size of the problem domain at the cluster level. During the allocation process, the bundles sent by nodes will decrease in size due to nodes not communicating tasks that they do not update (only tasks that they can win bids for are updated at each pass). This will cause a progressive drop off in bundle size as the network approaches consensus.

The complexity of communication can be expressed as a number of communication events required to achieve consensus. In networks comprised of a small number of nodes, communication is less complex, even for large numbers of tasks (assuming task lists remain short enough to stay within network interface MTU values). In CF-CBBA, this applies prominently to the cluster head level of allocation, where nodes must deal with the undivided problem domain. A larger number of clusters heads always causes a rise in the number of communication events, and thus bytes sent during consensus formation.

However, the number of cluster heads may cause small problem domains to be allocated to some clusters but not others, due to the respective fitness of one cluster over its peers. The result of this is a reduction in the data transmitted and communication events, as clusters without tasks do not participate in CF-CBBA. This can also be seen as a negative, however, as a network with clusters that are not servicing tasks can be seen as one that is over-resourced for the problem at hand. This, however, is considered to be a problem out of the scope of this research and an outcome of mission planning, not CF-CBBA.

Clustering of the network also has an effect on the optimality of task allocation. The number of nodes involved in task allocation has been shown to affect the optimality of assignment. A higher number of nodes allows for the optimal allocation of a higher number of tasks. This correlates with the number of cluster-heads in the initial phase of CF-CBBA. Sub-optimal task allocation in the outer loop results in sub-optimal allocation in the over-subscribed clusters. This can lead to a trade-off between optimality of task assignments and communication events in networks sub-divided into few clusters for large problems. This opens up options to administrators, allowing them to determine a minimum acceptable standard of optimality and configuring the network in such a way as to allow reduction of communication events within the confines of the optimality requirement that has been set.

CF-CBBA has been found to reduce communication cost and complexity, but the number of cluster heads must be chosen by the operator or generated automatically by the network itself, with reference to the size of the problem domain to prevent adverse effects on assignment optimality. By analysing the number of tasks likely to be processed in a task allocation procedure it may be possible to identify the appropriate manner in which to divide the network, but this is considered to be outside the scope of this research.

5.6.2 Analysis of BECF-CBBA

5.6.2.1 Number of Communication Events

Figure 5-10 shows the percentage of CBBA communication required to reach consensus under CF-CBBA, unclustered BECF-CBBA and clustered BECF-CBBA. All three approaches require significantly less communication than CBBA. CBBA is shown for reference.

CF-CBBA, in small networks such as the 18 node network analysed here, retains the gains shown in section 5.4.1, but to a lesser degree. Between 18% and 22% of the communication events required by CBBA is used, due to the division of the network into 6 clusters of 3 nodes.

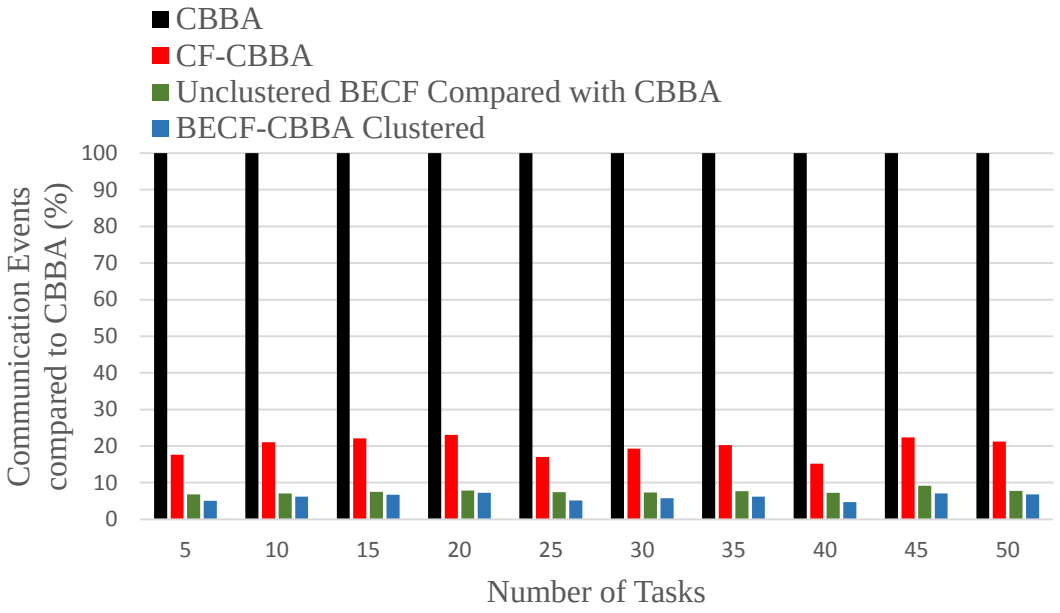

Figure 5-10 Chart comparing the difference in communication events between CBBA and CF-CBBA & BECF-CBBA

Clustered BECF-CBBA, due to its ability to multicast messages, significantly reduces the number of communication events required. Clustered BECF, in the same 6 cluster 3 node configuration as CF-CBBA, consistently provides the lowest number of communication events. Unclustered BECF-CBBA follows a similar trend, though the higher number of concurrent participants in communication drives up the communication event count relative to that recorded for clustered BECF-CBBA. Both approaches require less than 10% of the communication events required for CBBA. In the case of clustered BECF-CBBA, this value is consistently below 7% of the events required by CBBA.

Figure 5-11 compares BECF-CBBA communication events to those of CF-CBBA. Unclustered BECF-CBBA in the 18 node network requires between 6% and 9% of the communication needed for CBBA to arrive at a solution. As previously noted, the majority of these gains are driven by the reduction of redundant communication. By broadcasting instead of addressing nodes sequentially, the total number of communication events is significantly reduced.

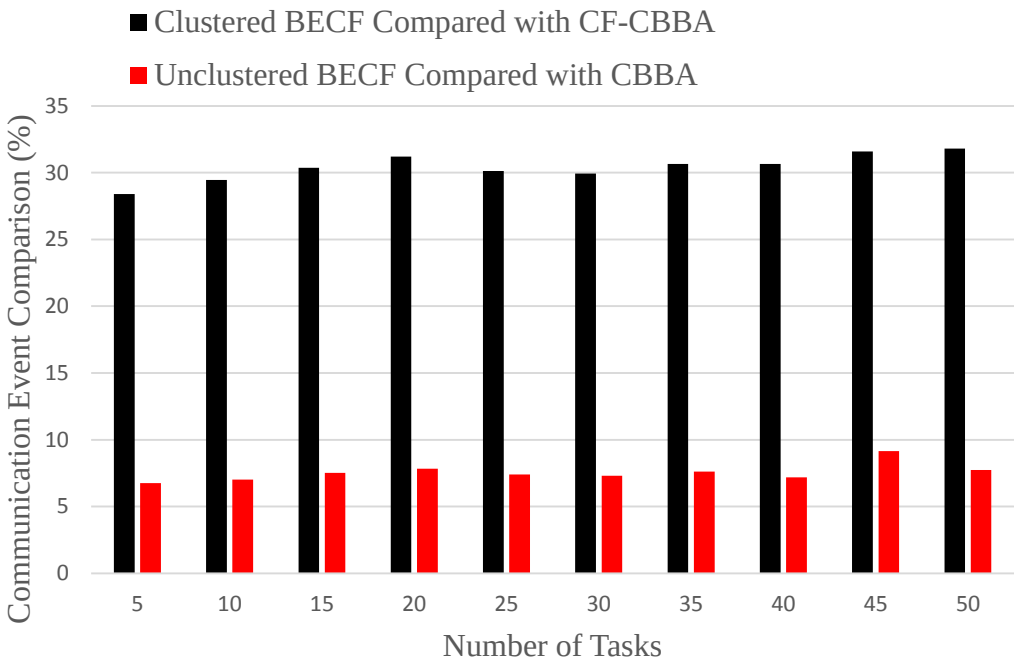

Figure 5-11 Chart showing the difference in communication events required between CF-CBBA and Clustered BECF-CBBA, and CBBA and Unclustered BECF-CBBA

Clustered BECF-CBBA has been compared with CF-CBBA to provide a foundation for analysing the effects of broadcast and multicast communication on clustered networks performing DTA. The difference between BECF-CBBA and CF-CBBA in a 6 cluster, 3 node per cluster configuration is less pronounced than that seen when comparing BECF-CBBA and CBBA in 18 node network, but is still significant. BECF-CBBA requires a maximum of 31% of the communication events needed by CF-CBBA, again displaying the effectiveness of cutting out redundant communication by exchanging bundles in a multicast manner. BECF-CBBA uses 28% of the communication required by CF-CBBA for 5 tasks problems, and 31% for 50 task problems.

In larger networks, these gains will increase, as both the outer and inner loops of BECF-CBBA task allocation benefit from the reduction of redundancy by utilising the multicast capabilities of wireless communication.

5.6.2.2 Number of Bytes Transmitted

Figure 5-12 shows a series of comparisons, showing the proportion of data sent by an algorithm when compared to its more data intensive counterpart.

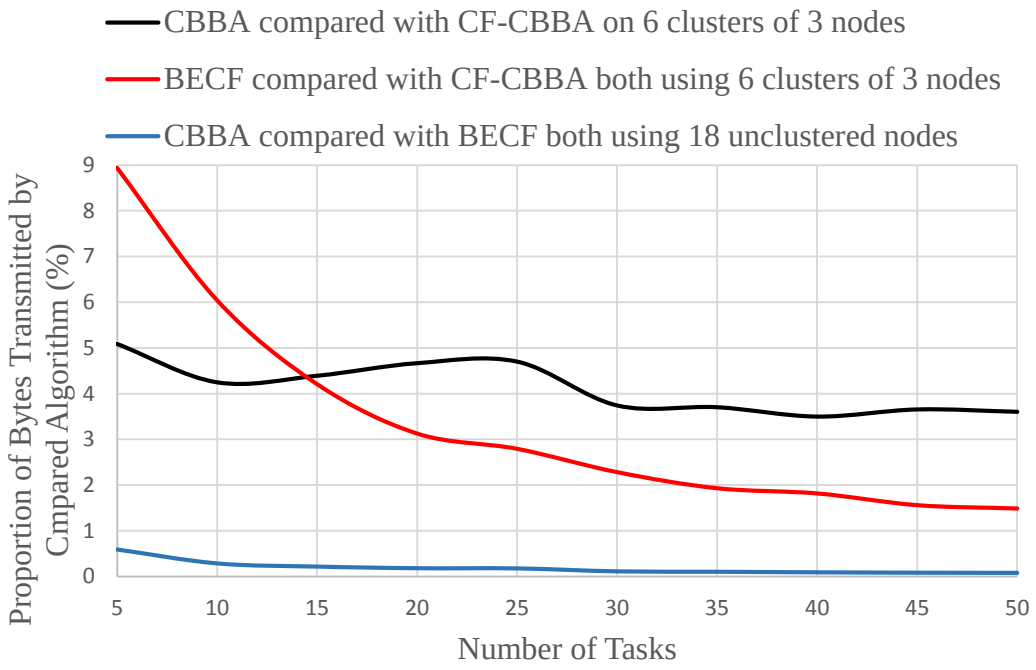

Figure 5-12 Graph showing the proportion of data sent by an algorithm, compared against its equivalent

CF-CBBA in a 6 clusters of 3 nodes configuration is compared with CBBA as a baseline for the effects of clustering on communication complexity and use of network resources under unicast communication. CF-CBBA in the same configuration is compared against BECF-CBBA, also in the same configuration, to show the effects of broadcast communication on clustered task allocation. This is followed by a comparison of unclustered BECF-CBBA and CBBA, to demonstrate the effects of broadcast communication in networks without clustering.

CBBA requires much more data to be sent to achieve consensus than CF-CBBA in either of the previously discussed configurations. CF-CBBA requires a maximum of 5% of the communication needed by CBBA, demonstrating that the clustering of the network has a significant impact on the amount of communication required. The amount of proportional

data required by CF-CBBA decreases with the size of the problem domain, reaching a low of 3.7% of the communication required by CBBA for 50 tasks.

BECF-CBBA in the same cluster configuration as CF-CBBA (6 clusters of 3 nodes) also shows a marked decrease in communication complexity and thus the volume of data required to achieve consensus. 9% of the communication required by CF-CBBA is needed by BECF-CBBA. This low number is driven by the removal of redundant communication from the network, BECF-CBBA, by broadcasting, does not repeat identical transmissions during the communication of bundles. This means that even with the addition of control packets to allow for the synchronisation of nodes during bundle sharing, BECF-CBBA requires far less network traffic to achieve consensus.

The cost of control is proportional to the number of nodes in the network, as synchronisation is required every time a bundle is shared to queue the next node to share their bundle. So as the problem domain increases in size, the cost of control is lowered, proportional to the data sent when communicating bundles. This results in a rapid decrease in the proportional amount of data required by BECF-CBBA when compared with CF-CBBA, leading to a low of 2.4% of the data required by CF-CBBA for 50 tasks. This clearly demonstrates the benefits of reducing redundant communication, even when control costs are incurred as a result.

BECF-CBBA, when compared against CBBA in an unclustered 18 node network, show significant gains. Both algorithms require more data to be sent than their respective clustered implementations, but the difference between BECF-CBBA and CBBA is significant. BECF-CBBA never requires more than 0.6% of the data transmission required by CBBA. This drops to 0.12% after 35 tasks. As the amount of redundant communication in CBBA is proportional to the number of nodes involved in the task allocation process, the removal of communication redundancy has a significant impact, reducing the required communication events and amount of data sent by up to 99.78% in these simulations.

5.6.2.3 Summary

BECF-CBBA has been compared with CBBA and CF-CBBA in a small network. This investigation has been conducted with a focus on the number of communication events required to achieve consensus.

The reduction of redundancy in communication has been shown to reduce the number of communication events. The reduction of communication events can be seen as the reduction of communication complexity. By reducing the number of messages that must be sent to achieve consensus, the number of packets sent may also be reduced, potentially reducing the impact of communication on network resources such as bandwidth and channel utilisation.

This is apparent in the results regarding BECF-CBBA's data transmission requirements, showing an above 99% reduction in the amount of network traffic required to achieve consensus, eve with the addition of control packets to synchronise nodes between bundle share broadcasts. Clustering the network provides more gains, further reducing communication complexity and network resource utilisation.

Broadcast communication dramatically decrease the number of communication events and the amount of data required to achieve consensus, even in unclustered networks of 18 nodes. BECF-CBBA further reduces communication complexity, beyond the reduction offered by CF-CBBA, without impairing mission optimality. This extends the number of feasible cluster configurations available to users when considering how best to compromise between optimality, speed and communication complexity.

5.7 Chapter Summary

A series of experiments have been designed to analyse the characteristics of CF-CBBA and BECF-CBBA when compared with CBBA. The results of those experiments have been shown and discussed, finding both proposed algorithms to deliver results with far fewer communication events.

Potential issues with assignment optimality have been identified, with analysis suggesting that the size of the network compared against the number of tasks to be allocated plays a large role in the optimality of the resulting assignment. This could potentially lead to the future proposal of appropriate fitness functions to assist in balancing the competing demanding of optimality and efficiency. Such a function could also be extended to account for cluster size inequalities. Section 4.5 highlights the potential for cluster size inequality to lead to sub-optimal task allocation, a hypothesis supported by the results shown in Section 5.4. The drop in optimality observed for smaller numbers of cluster heads (relative to number of tasks) would remain true for smaller clusters over-subscribed with tasks due to the assumption that all clusters are of equal size.

BECF-CBBA has been shown to spend the least time communicating. It also requires the lowest number of communication events. By efficiently utilising the wireless-medium, the network is capable of rapidly communicating task allocation problems with a minimum of redundancy.

6 SUPERMAN: A CLOSED-MANET SECURITY FRAMEWORK

6.1 Chapter Introduction

It has been identified in the previous chapters that autonomous MANETS have two critical requirements: a means by which nodes may perform tasks and pursue missions without human intervention, and a method of communicating changes in state and potential solutions. The reliability of these control and network services is paramount to the continued functioning of the MANET as an autonomous entity.

It is therefore vital that security is applied to both network and control services in an autonomous MANET, to ensure that it is not only capable of operating independent of human controllers, but able to remain reliable in potentially hostile situations. This is especially important in situations that have an element of hazard.

This chapter proposes the Security using Pre-Existing Routing protocols for Mobile Ad hoc Networks (SUPERMAN) framework, providing a full account of its inception, functionality and architecture. The need to secure network and control services is discussed and the SUPERMAN features relevant to these security considerations are detailed.

6.1.1 Chapter Layout

This chapter is laid out as follows:

- Section 6.2 provides a digest of the terminology used throughout the chapter, much of it unique to the proposed security framework.
- Section 6.3 outlines the research methodology used to derive the proposal that forms the core of this chapter from the gaps analysis and problem analysis that have been performed.

- Section 6.4 introduces the SUPERMAN framework and discusses the security services it provides.
- Section 6.5 summarises the chapter.

6.2 Terminology

Key terms used when describing SUPERMAN include:

- Trusted Authority (TA).
    - A static node responsible for node initialisation and provision of certificates; it is a prerequisite to SUPERMAN.
- Certificate public key (*CKp*).
    - Generated locally and shared with other nodes to join the network.
- Certificate private key (*CKs*).
    - Generated locally and kept private.
- Public Diffie-Hellman Key Share (*DKSp*).
    - A public value communicated as a part of Diffie-Hellman key exchange between nodes.
- Private Diffie-Hellman Key Share (*DKSpriv*).
    - A private value, held by all nodes in the network and never communicated. Used as the shared secret for Diffie-Hellman key exchange.
- Identifier (*I*).
    - A unique identifier, likely to be the Internet Protocol (IP) address in an IP-based network.
- Encrypted Payload (*EP*).
    - Payload data encrypted using an encryption scheme such as AEAD.
- Symmetric key (*SK*).
    - *SKe(s,d)* represents two nodes (source and destination) using a shared symmetric key for end-to-end security, derived locally via KDF from the Diffie-Hellman key generated by using DKSp and DKSpriv values.

- *SKp(s,d)* shared by two nodes (route neighbours); used to authenticate traffic as it moves along the network, derived locally via KDF from the Diffie-Hellman key generated by using DKSp and DKSpriv values.
- Symmetric broadcast key (*SKb*), shared with newcomer nodes by an authenticating node, generated by the first node to initialise the network. Differentiated into two application specific keys by a network-wide KDF stored locally on each node.
  - Symmetric end-to-end broadcast key (SKbe).
  - Symmetric point-to-point broadcast key (Skbp).

6.3 Research Methodology

The proposal and design of a security framework for MANETs requires the deconstruction of two hypotheses posed in Sub-section 3.4.2 into clear goals for the research. The first hypothesis considered as a foundation for this research is:

> *Enforcing rigorous access control policies on all nodes in a MANET will mitigate the open-medium problem.*

The open-medium problem is characterised by a vulnerability to passive attacks, such as those which eavesdrop on communication or record identifying information regarding nodes in the network. This can lead to attacks quickly identifying vectors for more active attacks, targeting the topology generation mechanisms (routing protocols) or control communication of the network to damage, delay or destroy the ability to provide key services.

A closed-MANET may provide the means to prevent trivial access to the network. The open-medium used for wireless communication, usually radio, is a persistent problem for network security as it is simple for attackers to passively glean information by observation. By preventing outsider nodes from accessing the network or being able to comprehend intercepted transmissions, they may be locked out of the network, providing a closed environment in which only member nodes may function.

Closing the network effectively will require the identification of critical areas of investigation. These will be identified by analysing the following vulnerabilities and potential counter-measures to the open-medium problem, building on the problem analysis provided in Chapter 3 and detailing the specifics of how each point may be addressed:

- Authentication of nodes must be analysed to identify ways of identifying legitimate nodes in the field.
- Access control must be investigated to determine low-cost mechanisms to prevent trivial access to network services such as routing and secure communication.
- Methods of providing confidentiality services in the network must be identified to prevent observation of identifying information and control communication by potentially malicious nodes.

These areas of investigation represent sub-goals when attempting to propose and design a closed-network MANET framework that will allow for further research into the network costs associated with such an approach to MANET security. Similarly, the concept of cost must be defined, but this is within the scope of the analysis of simulation results and will be more clearly defined in Chapter 7.

In addition to closing the network through the application of authentication and access control services, the data transmitted during communication must be protected. The following testing hypothesis outlines a method of reducing the cost of access control and authentication:

> *Allowing authenticated nodes to service authentication requests on behalf of their peers (if they share a route), will reduce the effective length of the route between the requesting node and the target node.*

It is vital that confidentiality, integrity and authenticity are maintained. The trusted intermediary must communicate the shared credentials securely, to prevent identity attacker or man-in-the-middle subversion of the process. The research required to propose such approaches can be broken down into the following sub-goals:

- Encryption techniques must be analysed to identify approaches to confidentiality that are appropriate for resource constrained wireless networks.

- Methods of providing integrity to data sent across the network must be identified, to allow the network to identify messages from legitimate nodes and deny the opportunity for malicious nodes to impersonate legitimate nodes using easily obtained identifying information.
- The application of confidentiality between end-points and on intermediate nodes along routes must be analysed to determine appropriate methods of providing security of both the data and the route it is transmitted over.
- Key generation and management approaches must be analysed to determine their suitability for use in resource constrained networks.

These goals represent the steps that must be taken to produce a proposed security approach that allows the hypotheses discussed previously to be tested. The last goal discussed above relates to both data security and closed-networks, requiring management of both the closure of the network and the management of security keys during the lifetime of the network.

The proposal of SUPERMAN is broken down into two sections, each directly related to the previously discussed hypotheses and research objectives:

- Closed-MANETs.
    - Access control services will be proposed to prevent trivial access to the network.
    - Authentication measures will be proposed to allow autonomous MANETs to allow nodes to join and leave the network securely, without human intervention or oversight.
- Data Security.
    - Routing security will be discussed and solutions to the problem of ensuring secure routing of information in autonomous MANETs proposed, relating to data security.
    - Communication security will be detailed, breaking it down in terms of data security. Methods to provide confidentiality and integrity to vital, non-network services will be proposed.

Data security will consider routing and control services. Both of these services require communication to perform their respective duties in the network, providing a topology and maintaining it or allowing for the optimal distribution of tasks throughout the network. Regardless of their primary function, services under these two categories require confidentiality and integrity services.

The proposal of appropriate methods of implementing these security services will be broken down into two sub-section: confidentiality and integrity. By identifying the requirements and means of providing these services, it will be possible to map them to one another and the previously discussed access and authentication services required to close the MANET.

6.4 SUPERMAN

Security Using Pre-Existing Routing protocols for MANETs (SUPERMAN) is a closed-MANET approach to security in mobile autonomous networks. The core purpose of the framework is to prevent trivial observation of network data and deny entry to unauthenticated nodes, while providing flexible and cost-effective security.

The highly changeable nature of MANETs can lead to complications in centrally administrated security, so a distributed approach is proposed, involving all nodes in the network in the administration and maintenance of security. This avoids complications that might arise from the loss of central controller nodes. Temporary or permanent loss of security administration would prevent the network from authenticating new nodes, undermining a main feature of the proposed framework.

6.4.1 Fundamental Concepts

To identify nodes as legitimate, security information is required. The identity of a node is includes its network address, security credentials and functionality. SUPERMAN is only required to consider identity-related information such as the address and security

credentials of a node for authentication purposes, making these two attributes the main focus when determining legitimacy of a node.

Security information can take a variety of forms. Private information, such as a common private key, is required to identify a node as a part of a network. Exchanging this vital information is a fundamental requirement of authentication. Nodes require information tying them to a common secret held by the network (and thus all nodes that are a part of it), and local secret information that allows security to be established end-to-end between specific nodes.

To provide a closed-network environment in the MANET paradigm, nodes must be secured individually, as there is no way to control the propagation of transmissions or direct attackers through specific *security* nodes in the same way as wireline networks can. This can be termed *node-by-node* security, where security is applied at the node level and due to the collective security of all nodes in the MANET, allows the network itself to be deemed secure. This can be described as a *virtual closed network*.

6.4.1.1 Virtual Closed Networks

Virtual closed networks are those which are closed by the collective enforcement of security in a uniform manner across all member nodes, instead of relying on security infrastructure to provide a safe space for the network. Wireline networks are capable of shielding internal elements of the network from outside interference by using hardware firewalls and similar systems. A MANET, however, is unable to guarantee the safety of nodes by enforcing gateways between the MANET and the outside world. The omnidirectional propagation of radio communication, for example, renders attempts to control the proliferation of communication meaningless without security measures applied to the communication itself.

Figure 6-1 shows a MANET of 12 nodes, all interconnected to varying degrees with other nodes. The lines represent links between nodes that are within communication range of each other. Any node with a link to another can receive communication sent by that node, even if it is not intended for it.

For example, if node 5 transmits a message to node 3, unintended nodes 6, 7 and 8 may listen to it. If the message is not encrypted, they will be able to observe the full contents of the message every time a message is sent. Taking this a step further, if 5 needs to communicate with 4, and routes over 3, the initial transmission is visible to unintended nodes 6, 7 and 8, and the retransmission is visible to unintended nodes 1 and 2. This exposes all communication to five unintended recipients, of which any may be malicious without authentication measures to prove otherwise.

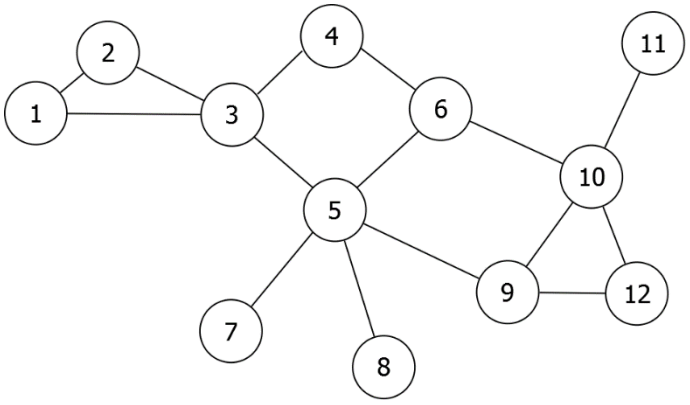

Figure 6-1 Diagram showing a MANET of 12 nodes

To protect the MANET from passive observation and the attacks that may be possible acting on observed and derived information, the communication itself must be secured. The lowest unit of communication hardware in a MANET is the node, heterogeneity in MANETs in terms of node communication capability extends few benefits to security due to the aforementioned omnidirectional propagation problem.

Figure 6-2 illustrates an abstraction of a possible approach to closing a MANET. Nodes 1 and 2 are outside of the network, and are not members of the closed-MANET. All blue nodes are members of the MANET and have authenticated with the network and each other. They apply confidentiality services to outbound communication and perform integrity checks on inbound communication. This provides a means of identifying the source of a message and ensuring that messages are from the nodes that are claimed to have sent them. It also prevents trivial observation of communication.

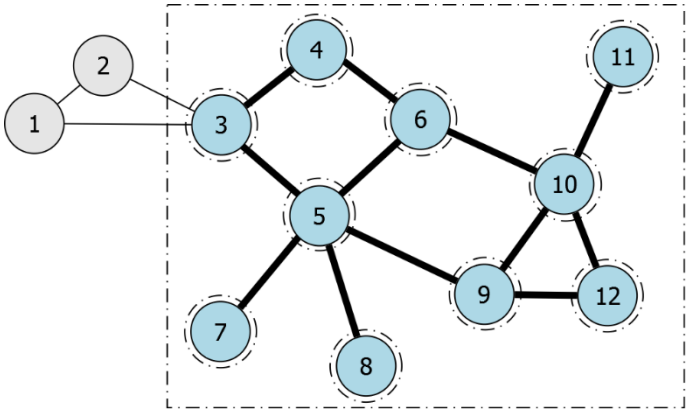

Figure 6-2 Diagram showing a SUPERMAN VCN of 10 nodes

The dashed line encompassing the closed MANET is an abstraction. It shows that any nodes wishing to communicate with nodes inside the box must participate in, and pass, authentication checks which allow access control policies to be enforced. If node 2 attempted to communicate with node 3, node 3 would reply with a challenge to provide valid credentials. Failure to provide these credentials results in node 2 remaining outside of the closed network.

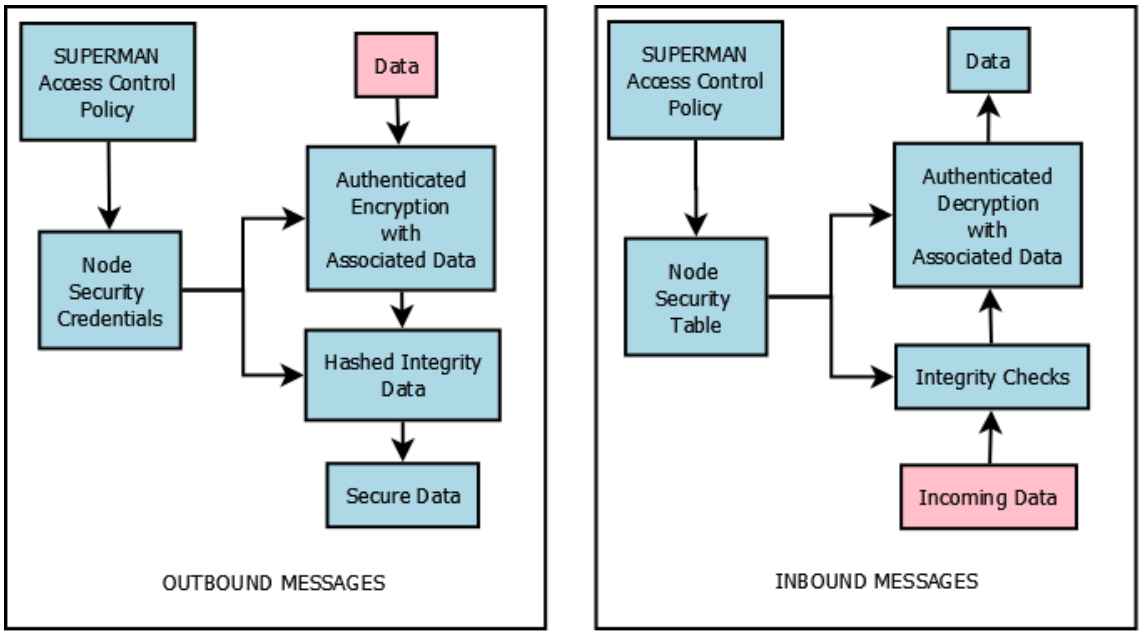

Figure 6-3 Diagram showing the protection of outbound messages and the closure of the network to unauthenticated incoming messages

The dashed circles around each node represent the real closed elements of the network. By ensuring that access control policy and security services are enforced in a global manner across all nodes, the individual closure of a node to outside observation and attack may collectively provide the abstract closure of the whole network. Figure 6-3 shows the enforcement of security on a node during the transmission and reception of a packet, identifying the critical security services represented by the dashed circles.

Outbound messages are afforded protection by encrypting packets using keys unique to the nodes involved in the communication, providing authenticatable encryption (encryption that provides both confidentiality and authentication services). The SUPERMAN access control policy dictates whether a target node is legitimate or not. Nodes identified as legitimate may be communicated with following the security policies in place. All unknown nodes are challenged to prove their legitimacy prior to further communication. Should the node be a legitimate member of the network, the encrypted packet is digitally signed prior to transmission to allow the packet's integrity to be checked by the next node in the route, and to allow the identity of each subsequent node to be determined by those that receive the relayed packet as a precaution against masquerade attacks.

Inbound packets are integrity checked at layer two to determine if the packet has been corrupted during transit. The identity of the last node to relay the packet may also be checked for validity by encrypting the digest of the packet payload (hash) with the SKp of the relaying node. Assuming the packet passes these checks, authenticated decryption may be performed to reveal the contents of the packet and provide identity checks against the source node. This is vital to ensure that the originator of the message is a valid member of the network, and that end-to-end authentication policy is enforced. Once decrypted and found to be from a legitimate node, the data may then be passed up the stack for further processing.

The result of this framework may be referred to as a *virtual closed network* as the closure of the network is an abstract outcome of the real closure of all constituent nodes.

6.4.1.2 Security Services and Modes of Operation

SUPERMAN has two main phases of operation: an initialisation phase which occurs at a base station, and a deployed phase. The second phase assumes zero communication with a base due to a combination of feasibility (range, obstruction of the communication medium by terrain) and security (long range communication to base gives sophisticated attackers the opportunity to identify and locate bases of operation).

Data security is vital to prevent the snooping of network information. As packets may always be captured by passive observers when using wireless communication, the obfuscation of packet contents is a critical consideration. Confidentiality services prevent attackers from easily reading packets by encrypting the contents, so that only nodes with the correct cryptographic key can read them. Attacks on cryptographic keys are non-trivial, requiring significant computational power and a large sample from which to derive key information.

6.4.2 SUPERMAN Framework Overview

SUPERMAN is comprised of the following elements:

- Architecture.
    - Defining the position of SUPERMAN in the network stack, using the OSI model as a template.
- Certificate format.
    - Describing the information required in a certificate and the security applied to prevent the misuse of security credentials.
- Packet types.
    - The bespoke packet types required by SUPERMAN are detailed, with reference to the processes that they are used by.
- Security table.
    - The format of the security table used by every SUPERMAN node will be defined, with discussion of the information stored in it.

6.4.2.1 Architecture

SUPERMAN operates at the network layer of the OSI model, interacting with all outbound and inbound IP packets on each node. This framework is not intended to interact with the functioning of network services such as routing, instead focusing on the security of the data being sent and received by such services. As a result, SUPERMAN plays no role in the definition or maintenance of the topology of the network, but does provide security to those services.

Figure 6-4 shows the services offered by SUPERMAN and their relationship with other network services. SUPERMAN has no direct interaction with the stack outside of the network layer.

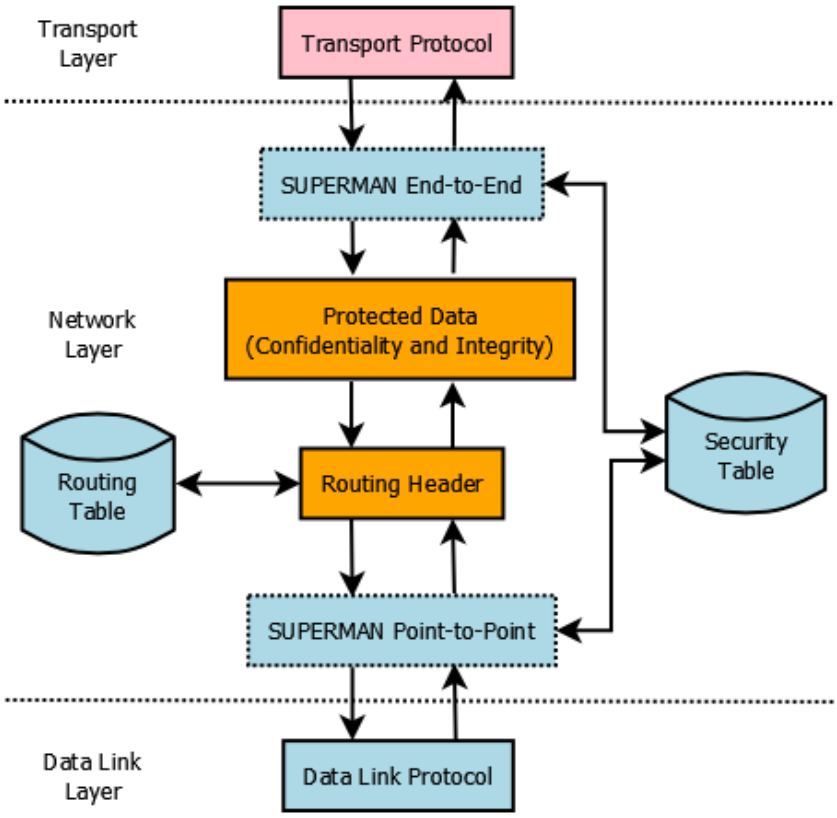

Figure 6-4 Diagram to illustrate the additional SUPERMAN encryption and authentication services (indicated by dashed outlines)

The payload type for each incoming IP packet is checked at the network layer. All incoming packets not containing SUPERMAN payloads are dropped. The security of the packet, including authenticated integrity and confidentiality, is then checked. The routing protocol used to construct and maintain the network topology determines whether packets are advanced further up the stack or relayed to the next node in a route. This occurs both end-to-end and point-to-point at each node in a route until the destination is reached.

Outbound packets are not subject to this process, passing through end-to-end encryption and point-to-point digital signing regardless of the length of the route between source and destination. All that matters is that the appropriate end-to-end cryptographic key is used, and a signature that can be authenticated by the next node on the route is appended.

A security table is required to facilitate the security process. The security table contains the node identity (I) for a given node, DKSp, SK(s,d) and SKp(s,d). This data provides the credentials required to authenticate messages from nodes with an entry in the table. The length of a route is not required for incoming or outgoing packets, only the identity of the source and destination nodes (for end-to-end communication) and the source and next intermediate node (for point-to-point communication). Furthermore, the routing protocol used is irrelevant to the functioning of SUPERMAN. If the target node(s) have entries in the security table, SUPERMAN applies the appropriate encryption and signature for outgoing packets, and integrity checking and decryption for incoming packets.

If either node does not have a valid security table entry, then the authentication and access control processes are initiated to allow the sending or receiving node to associate with the unknown node(s).

Outgoing packets, once they have been encrypted and signed, are sent to the data-link layer for encapsulation in an appropriate packet, such as an 802.11 packet. Incoming packets are passed up the stack via the transport layer after integrity checks return valid neighbouring node identity and the packet successfully decrypts using the key formed between the destination and source node. Any packets failing to meet the above criteria are dropped. SUPERMAN does not incorporate trust or reputation systems, though it is able to provide data regarding failure to pass security checks, should such a system be desired.

6.4.2.2 Certificate Format

Certificates provide a means of encapsulating identifying data and security credentials for exchange with other nodes. By presenting node-identities as certificates, the required variables may be presented in a single array of values.

Certificate-based representation of node identities for the purpose of authentication also allows certificates to be signed by a Trusted Authority (TA), which can assign certificates prior to a mission and is not required to be involved in certificate exchange in the field. By digitally signing certificates, integrity is applied, allowing nodes to check if the certificates they receive are legitimate and have been authorised by the TA. By associating node identities with a TA, node legitimacy can be ascertained through authentication of the node and its relationship with the TA through holding a valid certificate.

6.4.2.3 Packet Types

Every SUPERMAN packet uses a common header, shown in Figure 6-5. The SUPERMAN Header (SH) is divided into three variables: packet type, a timestamp (or other appropriate unique number such as a sequence number), and the length of the payload to follow. SUPERMAN Headers are 5 bytes in length.

Octets	0	1	2	3	4
0	Type	Timestamp		Protocol Identifier	

Figure 6-5 SUPERMAN Packet Header (SH)

Time stamping provides the quality of uniqueness to transmitted packets. It is this uniqueness that can be used to determine if a packet has been replayed, mitigating the effectiveness of replay attacks. By using the current time, or another unique local variable,

to timestamp outbound packets, a node is able to identify recurring values. Recurrence may indicate that a packet has been replayed, allowing the node to discard it after a match is found with recently received packets. By discarding such packets as soon as possible, the impact of replay attacks may be reduced.

The protocol identifier defines the type of incoming packet after the SUPERMAN Header and before any footer appended by the SUPERMAN framework. This allows for the identification of the transport layer protocol expected (such as TCP or UDP) when the packet is escalated out of the network layer.

Payloads will be encrypted, and so must be subject to decryption processes before being escalated to higher levels of the network stack. Footers contain integrity assurance information, which must be checked against a digest of the packet. As the length of a SUPERMAN packet is equal to the payload (dynamic length) plus the header and footer, the length of a packet can be calculated locally, without the need for a length value in the header.

Table 6-1 SUPERMAN Packet Sizes

ID	Type ID	Packet Type	Size (Bytes)
01	DReq	Discovery Request	SH+DKSp
02	CReq	Certificate Request	SH+DKSp
03	CEx	Certificate Exchange	SH+CKp+Sa
04	CExB	Certificate Exchange with Broadcast Key	SH+CKp+SKb +Sa
05	DKSpReq	DKSp Request	SH+S
06	DKSpRes	DKSp Response	SH+DKSp+S
07	SKI	SK Invalidation	SH+I+S
08	BEx	Broadcast Key Exchange	SH+SKb+S
09	DP	Data Packet	SH+EP+S

The type of packet determines the processes that the packet will be subject to upon being received by the destination node(s). Table 6-1 shows the types of packets, their short form designations and the formulas for deriving their size in bytes.

Discovery Requests (DReq) and Certificate Requests (CReq) are very simple packets, comprised of a SUPERMAN Header with the appropriate type identified and a payload containing the node's Diffie-Hellman public Key Share (DKSp). These packets are not

encrypted, nor are they digitally signed, and are repeatedly broadcast by nodes that have not joined a network every few seconds in an attempt to establish contact with nodes that may grant them network access.

DReqs are simple requests for communication, which are required only to have a unique timestamp value to mitigate the effects of abuse through replay attacks. These packets also facilitate immediate symmetric key generation, allowing all further communication between those nodes to be encrypted, including all certificate exchange functions. This allows the process of joining the network to be undertaken confidentially. DReq packets also facilitate network merge, with nodes periodically broadcasting DReq packets after joining a network, at a much reduced poll rate defined by the network administrator.

Certificate Exchange (CEx) packets are used to exchange identity and security information between nodes. This information is used as part of the authentication and access control process.

Figure 6-6 shows a SUPERMAN CEx packet. It has a SUPERMAN Header, a Certificate (CKp) and a footer (Sa). The certificate is given to a node by the Trusted Authority upon initialisation, providing it with a means of identifying itself as a legitimate node to others that were initialised by that TA. The packet is digitally signed to allow the integrity of the packet to be checked, and to allow authentication of the packet against the TA which the source node claims to have been initialised by.

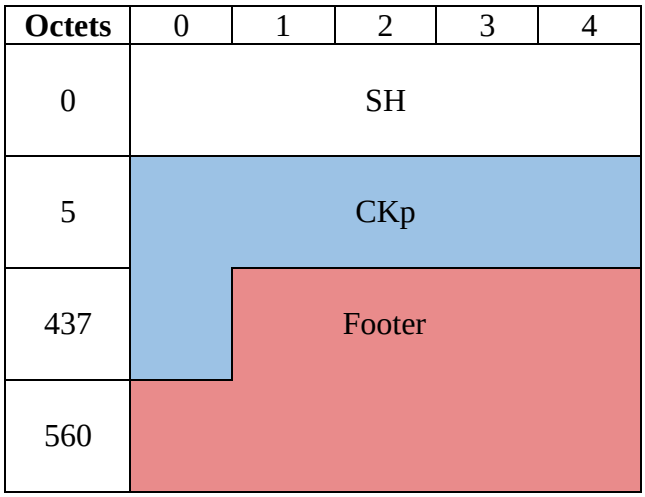

Figure 6-6 Example of a SUPERMAN Certificate Exchange Packet

Certificate Exchange with Broadcast Key (CExB) packets are sent in response to valid CEx packets. These contain the network broadcast key (SKb) for broadcast communication with other legitimate nodes, as well as the certificate held by the sending node. After certificates have been exchanged both ways, the querying node is considered to be a legitimate member of the network, and is no longer required to share its certificate.

Nodes that are already members of the network use DKSp Request (DKSpReq) and Response (DKspRep) packets to form associations between nodes. These are lightweight when compared with CEx and CExB packets, as they do not contain the certificate. The node has been authenticated and allowed access to the network under these conditions, granting it a network broadcast key share (SKb) from which end-to-end and point-to-point broadcast keys may be derived through the use of a Key Derivation Function (KDF).

Security Key Invalidation (SKI) packets are used to inform other nodes of the departure of a node, or in the case of implementations making use of trust or reputation systems, the demotion of a node to a de-authenticated state. The identity of the node is placed into the payload, and the packet is encrypted and initially signed using the source node's security keys.

Broadcast Key Exchange (BEx) packets are used to exchange broadcast keys where network merges are required. They may also be used to propagate any key change throughout the network, secured using the key to be discarded upon implementing the key in the packet payload.

Data (DP) packets are simple message packets. They allow non-SUPERMAN messages and data to be secured, providing the security services of the framework to all other communication on the network. Their payloads are dictated by the content required by the service that needs to communicate. Their payloads are encrypted and all packets are initially signed by the source node.

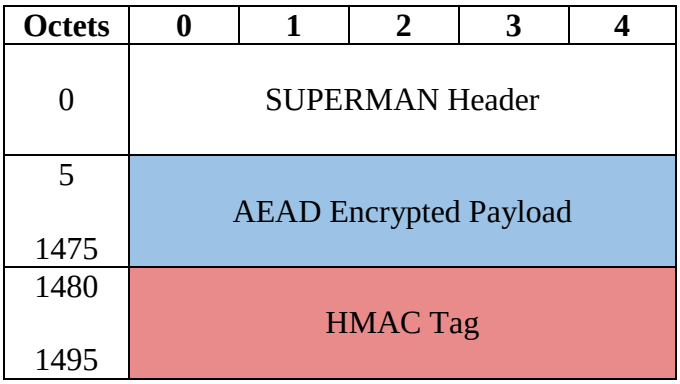

Figure 6-7 Example of a SUPERMAN Packet using AEAD and HMAC

SUPERMAN packets, except for DReq, CReq and CEx/CExB, use the common SUPERMAN packet format shown in Figure 6-7. All such packets are only able to be used by nodes that have joined the network and have the appropriate security associations and keys. Payloads may vary in size and content, but are all encrypted using Authenticated Encryption with Associated Data (AEAD) to provide confidentiality. Signatures are provided by use of Hash Message Authentication Codes (HMAC) for use in integrity and authentication checks.

6.4.2.4 Security Table

Every node in a SUPERMAN network possesses a security table. These tables represent their associations with other nodes in the network, and the keys that they share. An example of a SUPERMAN security table for a single node (a), is shown in Table 6-2.

Table 6-2 SUPERMAN Security Table

Node ID	SKe	SKp	DKSp	SKbe	SKbp
I(x)	SKe(a,x)	SKp(a,x)	DKSp(x)	-	-
I(y)	SKe(a,y)	SKp(a,y)	DKSp(y)	-	-
*	-	-	-	SKbe(n)	Skbp(n)

In this table, node A has formed associations with nodes x and y. The * entry represents a local entry for network wide data (SKbe(n) and SKbp(n)). In this case, n represents the network as an entity possessing end-to-end and point-to-point security keys used for broadcast communication. This entry is updated if a network merge occurs, or if any other key changing event occurs.

Security associations under SUPERMAN involve the exchange of certificates and DKSp information, which is stored in the security table to form an authenticated identity for the node. The exchange of certificates is performed when joining the network, but certificates (once validated) do not need to be stored and are not entered into the security table. DKSp information is stored to allow for key derivation and to allow for nodes to provide DKSp on behalf of other authenticated nodes during referred security association.

For every node associated with A, a pair of security keys is formed. These are both derived from the DKSp values communicated during association with a node. Through the use of Diffie-Hellman, a pairwise symmetric key is formed between two nodes in possession of each other's DKSp values. This key is derived into two keys via KDF, an end-to-end key (SKe) and a point-to-point key (SKp). End-to-end keys are used for cryptographic processes while point-to-point keys are used for the creation of HMAC tags, which form the footer of SUPERMAN packets.

Broadcast packets are encrypted using keys common to the network, derived via KDF from the SKb generated in new networks on first authentication of another node, and shared with nodes after they have joined the network. These keys are differentiated into end-to-end (SKbe) and point-to-point (SKbp) keys used in the same way as those derived from the DKSp, but for broadcast and multicast communication.

6.4.3 Crypto-key Structure

Each SUPERMAN node possesses keys associated with other nodes in the network, holding two keys for each node it has associated with. Each node will possesses four types of network key. These are:

- *SKe*, one for every associated node, used for end-to-end security.
- *SKp*, one for every associated node, used for point-to-point security.
- *SKbe*, one for the network, used for broadcast end-to-end security.
- *SKbp*, one for the network, used for point-to-point security.

SK values are generated using Diffie-Hellman and *SKe* and *SKp* keys derived from the *SK* value, which requires that all nodes possess:

- *DKSp*, a publicly communicated local value used in key generation.
- *DKSpriv*, a privately held global value used in key generation.
- Two global *differentiation* values, to allow the derivation of end-to-end and point-to-point keys.
- The global value *I*, which represented the number of iterations the KDF should perform.

During the mutual authentication process outlined in Sub-section 6.4.3.2, DKSp values are exchanged, allowing the following process:

1. Node *A* and node *B* are initialised by TA with modulus (*M*) and base (*g*) values that is common to all nodes. These variables form the *DKSpriv* value.
2. Each node generates a local value (*A* generates *a*, *B* generates *b*), the *DKSp* seed value.
3. Node *A* sends its *DKSp* value, $A = g^a \bmod M$, to node *B*.
4. Node *B* sends its *DKSp* value, $B = g^b \bmod M$, to node *A*.
5. Both nodes store the received *DKSp* values in their security tables.
6. Both nodes locally compute a solution (*SK*);
    a. Node *A* computes $SK = B^a \bmod M$.
    b. Node *B* computes $SK = A^b \bmod M$.
7. Both nodes now possess the solution *SK*, which is a value unique to the link between the two nodes that participated in the key generation process, the *SKe* and *SKp* keys are derived from this value using a KDF.

Equation 6-1 demonstrates how *SKe* and *SKp* values are derived. The values *e* and *p* represent the values used for end-to-end and point-to-point keys, respectively.

$$\mathbf{SKe} = \mathbf{KDF}(\mathbf{SK}, \mathbf{e}, \mathbf{I})$$

(Equation 6-1)

$$\mathbf{SKp} = \mathbf{KDF}(\mathbf{SK}, \mathbf{p}, \mathbf{I})$$

Due to the globally constant differentiation (*e* and *p)* values instead of a salt, dictionary attacks are possible if the differentiation values become known. As a result, nodes are not permitted to communicate differentiation values at any point. They are held locally and privately, in the same way as private keys and *DKSpriv* values.

SKbe and *SKbp* keys are randomly generated at point of initialisation, and given to each node by the network TA. If a node that does not have a broadcast key is authorised to join a SUPERMAN network, the existing broadcast key will be shared with the new node.

At no point does SUPERMAN allow for more than two nodes to exchange keys at a given time. Multi-party Diffie-Hellman is not considered within the context of this research, keys must be unique to the bidirectional link shared between two nodes.

When two networks need to cooperate, a gateway (node or virtual), or mutually-shared broadcast key must be established. This scenario is outside the scope of the research, but will be considered as an item of future work to expand the capabilities of SUPERMAN to multi-MANET and multi-network cooperative security.

6.4.4 Access Control and Authentication Processes

To provide a virtual closed network, SUPERMAN must enforce an access control policy and authentication services. These can be considered the core of the SUPERMAN framework, providing the means by which node identities may be reinforced with security information, and how that information allows the secure communication of information within the virtually closed network.

6.4.4.1 Initialising the Network

As discussed in Sub-section 6.4.1, SUPERMAN virtually closes the network by enforcing globally standardised security on every individual node in the network. Legitimate nodes are all initialised by a TA, which is responsible for providing a certificate and security policy information to the node prior to its deployment to the mission area.

At initialisation, each node generates a certificate, signed by the TA that acts as an authority for that network. This allows the certificate to be bound to the identity it represents, with the relationship between node and certificate being authenticated and signed by the TA to provide proof of origin for all other nodes that may receive the certificate as proof of legitimacy. This, by proxy, reinforces the relationship between the node and the TA. As the association of the node with the TA for the network is a critical element of proving its legitimacy, it is vital that these measures are enforced.

When a node has been initialised, it leaves for the mission area and is assumed to be potentially beyond the communications range of the TA. The formation of a network begins when nodes send discovery requests. The network itself is an abstract concept until this point, existing as the potential for secure communication between SUPERMAN nodes via initial certificate exchange. A node moving to the mission area will broadcast DReq packets in an attempt to contact other SUPERMAN nodes, until it receives a reply.

6.4.4.2 Joining the Network

To form the network, nodes must join it by authenticating with other nodes and therefore the network itself. By abiding by the security policies of the framework, nodes may share certificates, identify nodes from the same TA and form virtual closed networks with each other. This is a non-deterministic process; nodes will initialise this process upon contact.

A node broadcasting DReq packets will continue doing so until a CReq is received as a challenge from a SUPERMAN node. This initiates the network joining process. This process is a combination of the follow:

- Network access.
    - Allowing the node to join the network and providing an SKb to allow the generation of SKbe and SKbp via KDF. The node will also associate with the node granting access to the network during this process, resulting in all relevant security table entries for both nodes.
- Security association.
    - By authenticating with the newcomer, the authenticating node engages in a bidirectional exchange of DKSp values. This equips both nodes with the means to generate shared symmetric keys using the Diffie-Hellman key generation algorithm. As a result, the nodes can be said to be *associated* with one another from this point.

Figure 6-8 shows the communication process involved in the initial authentication of a newcomer node.

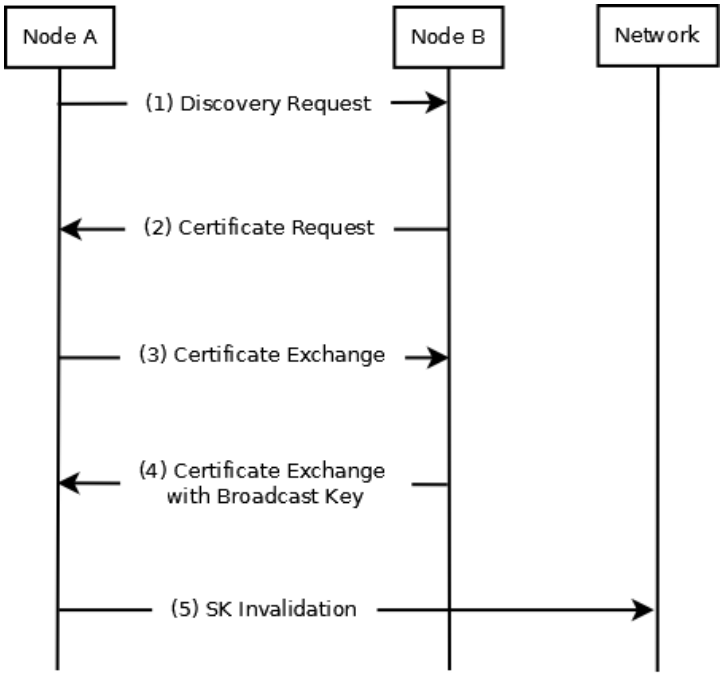

Figure 6-8 Sequence diagram to demonstrate the certificate exchange process

A is a node attempting to join the network and B is the first recipient of a DReq from A, and thus the first node to challenge A to prove its legitimacy. This process is bidirectional, A will respond to a challenge with a CEx packet to provide node B with its certificate, to prove that it has been initialised by the appropriate trusted authority (TA).

0. Each node is provided with a certificate from a TA, in order for it to become networkable.
1. The joining node (*A*) seeks to join a network by periodically broadcasting Discovery Request (*DReq*) packets containing its DKSp. This continues until it receives a Certificate Request (*CReq*) from a networkable node (*B*).
2. Having received a DReq from node A, node B sends a CReq packet containing its DKSp to A. Both nodes perform Diffie-Hellman using the shared DKSps they now hold, to generate SKe and SKp keys which are used to encrypt and provide integrity to the rest of the access control process.
3. Upon receiving a *CReq* from B:
    a. *A* sends its certificate in a Certificate Exchange (*CEx*) packet to *B*. *A* then sends a *CReq* to *B*.
    b. *B* checks the integrity and authenticity of the *CEx* packet, using the shared SKp.
    c. *B* checks the certificate's authenticity against the TA hierarchy of its own certificate. If the certificate is deemed authentic the node address, *CKp* is added to *B's* security table. If the certificate fails this check, the DKSp, SKe and SKp credentials generated for node A by B are dropped and B ceases the access control process silently.
4. *B* responds to *A's CEx* with its own *CExB*. *A* repeats steps *a* to *d* in 2. The CExB also provides A with an SKb, from which it may derive an SKbe and SKbp for broadcast communication, using the network KDF. *B* and *A* both invalidate any prior security associations they have with each other when receiving CEx or CExB packets with new information. This involves purging all previous information from their local security table entries for each other.
5. If either node holds previous security information about the other node at the end of this process, that information in invalidated and removed from their security tables as discussed in stage 4. This is followed by a broadcast SKI packet, invalidating those credentials, propagated to all node in the secure network to allow them to discard such information as well. This prevents the accumulation of expired security data on nodes that may be isolated from the initial invalidation event.

If *B* is the first node on the network, it would generate a broadcast key. This key is included in the CExB packet and sent with B's certificate and DKSp. The broadcast key

is encrypted with the SKe key derived from the SK for the link between A and B, which is itself the result of Diffie-Hellman key exchange using the private DKS held by node B and the public DKS it has received from node A. Node A may decrypt this portion of the packet once it has performed the same key generation procedure and used the KDF to generate an SKe and SKp to represent its secure association with B. Authenticated nodes may pass this on to any node requesting authentication with the network if they meet the above conditions.

At the end of this process, the newcomer node has authenticated with the legitimate node and joined the network. Both nodes share a symmetric key for point-to-point and end-to-end communication security. If this is the first event of its kind in the mission area, the network is formed at this point.

It is possible that multiple networks will form due to the distribution of nodes throughout the mission area, in which case SKb resolution is required. This also requires the exchange of certificates to check that the responding node is a part of the same network, or if its network is a part of a hierarchy of affiliated networks. SUPERMAN networks can be set to periodically broadcast merge requests using the existing DReq packets. The node sending DReqs will, upon completing the network authentication procedure, broadcast the new broadcast key to all members of its network, invalidating the current broadcast key and completing the network merge.

6.4.4.3 Delegated Authentication

SUPERMAN is intended to reduce the amount of communication required for authentication of nodes while retaining a high level of security. This may be achieved by allowing nodes to vouch for nodes which they have found to be legitimate through direct authentication as described in Sub-section 6.4.3.2, or by reference, as shown in Figure 6-9.

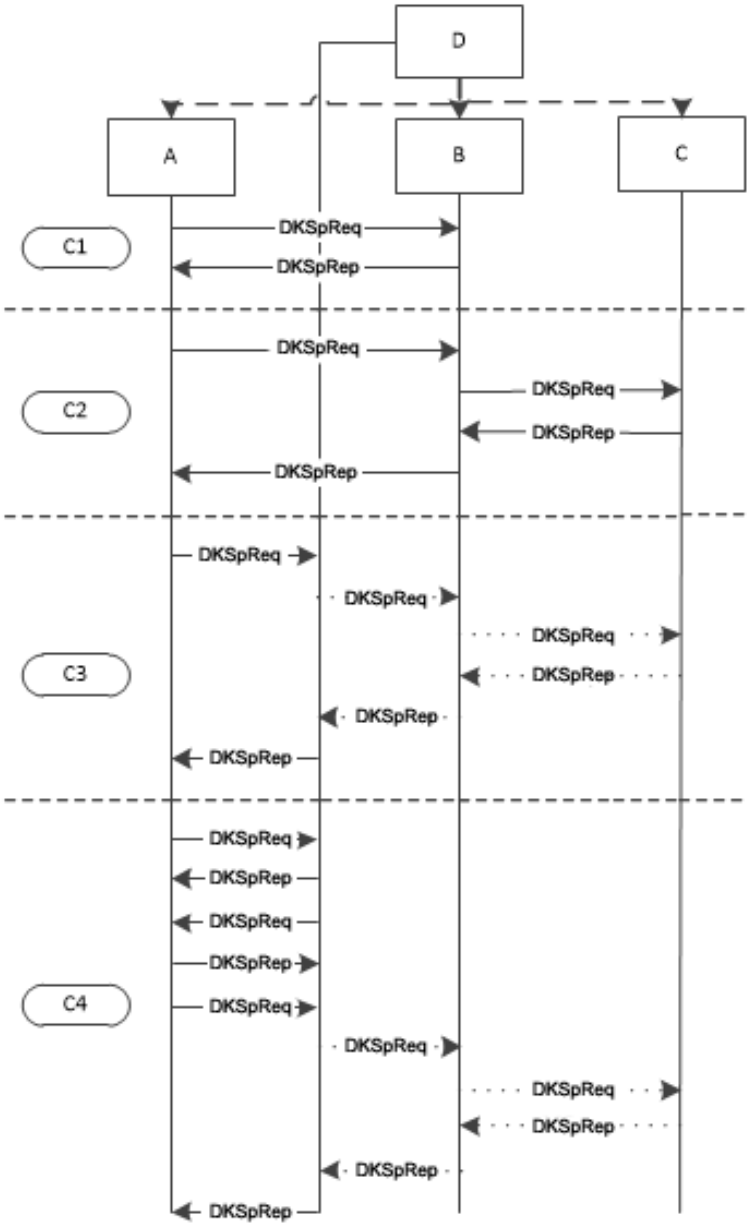

Figure 6-9 Sequence diagram to show Security Credential propagation after Network authentication

C1. *A* needs to communicate with *B* and A's adjacent to *B*. *A* lacks *B's* DKSp.

a. A *DKSpReq* is sent to *B* by *A*.

b. *B* responds with a *DKSpRep* containing its DKSp. *A* adds *B's DKSp* to its security table.

C2. *A* needs to communicate with *C* and requires an intermediate node *B* to relay communication. *A* and *B* do not know *C*, but know each other.

a. *A* sends a *DKSpReq* to *C* via *B*.

b. *C* is not known to *B*. *B* forwards the *DKSpReq* to *C*.

c. *C* replies to *B* with a *DKSpRep*.

d. *B* adds *C's DKSp* to its security table the forwards it on to *A*.

e. *A* receives *B's* forwarded *DKSpRep*, then adds *C's* security details to its security table.

C3. *A* needs to communicate with *C* and requires a route through *D* and *B* to reach *C*. *A* knows *D* but not *B* or *C*.

a. If nodes *D* or *B* hold the *DKSp* for *C*, they may respond on *C's* behalf and pass *C's* details on to *A* without ever contacting *C*. The dotted lines in Figure 5 represent optional communication that will not occur if a previous node holds *C's* security details.

C4. *A* needs to communicate with *C* but does not know *D or B.*

a. To send messages securely, *A* needs to know *D* and *C*. *A* will send a *DKSpReq* to *D*. *D* and *A* will associate with each other as per case 1.

b. When associated with *D, A* will send a *DKSpReq* addressed to *C*. *D* will relay the *DKSpReq* unless it has *C* in its security table.

c. If *D* does not have *C* in its security table, the procedure outlined in case 3 will be followed.

The end result of any of these scenarios is the authentication of the unknown nodes with node A, and the formation of security associations between node A, the target node and in the case of an unknown neighbouring node on the route, that node. In SUPERMAN networks with a high number of mutually authenticated nodes, the communication required to obtain the security credentials of a locally unknown node may be significantly reduced due to the vouching process allowing for the closest knowledgeable node to respond on the target's behalf.

6.4.5 Authentication, Confidentiality and Integrity Services

Authentication, confidentiality and integrity are the three security services that provide a virtual closed network environment. By ensuring that only authorised nodes may decipher messages, confidentiality within the network, and between nodes, may be ensured. Checking the integrity of packets removes the threat of man-in-the-middle attacks and other forms of malicious network activity that may modify or corrupt the contents of packets.

It is vital that communication in the network is reliable, trustworthy and obfuscated against meaningful observation. Reliability is provided by reducing the time spent processing potentially malicious packets, discarding packets that fail authentication checks as quickly as possible to reduce their impact on the receiving node's ability to service network traffic. Trustworthiness may be provided by verifying the identity of the source and last relay nodes, to prove that the packet is being sent by the legitimate nodes it claims to have been sent by. Obfuscation against meaningful observation involves encrypting the packet payload so that only header information remains trivially visible. By ensuring that packet contents cannot be read without attacks on the cryptography used by the network, it is possible to deny passive attackers the ability to derive vital network information.

These services are applied to all data packets in the network; control and network. Routing packets and DTA packets are all considered to be BP or DP packets in a SUPERMAN network, extending the security services of a virtual closed network to those services. As a result, routing may be deemed secure by virtue of routing packets being encrypted to provide confidentiality and the propagation of route requests (in the case of reactive routing protocols) being afforded authentication and integrity at each hop along a potential route.

In the same way, DTA communication is afforded confidentiality and integrity for all nodes which pass authentication. Those who do not may not participate in the network and are not considered to be members of the network for the purposes of SUPERMAN or any DTA processes.

6.4.5.1 End-to-end Confidentiality, Integrity and Authentication

Figure 6-10 shows an instance of communication between two legitimate nodes, A and B. The node C is a malicious node, seeking to launch a man-in-the-middle attack against the two nodes.

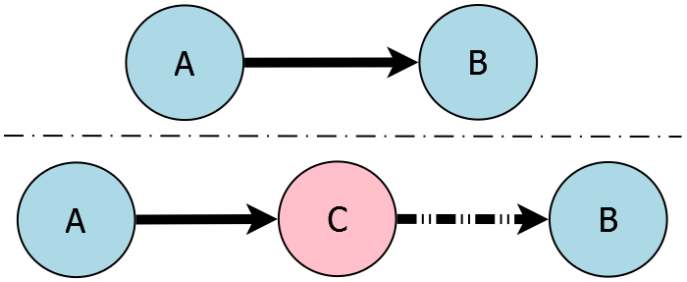

Figure 6-10 Diagram showing the modification of a packet by a man-in-the-middle attack

The purpose of this attack is to change the contents of packets sent by A to B and vice versa. Without confidentiality, integrity or authentication measures, it is a matter of receiving the packet before B and retransmitting the modified packet to pre-empt any communication between A and B with that of C.

Authenticated Encryption with Associated Data (AEAD) is used to provide end-to-end confidentiality and encryption services in the network. The purpose of this service is to provide source authentication and confidentiality in an end-to-end manner. As such the encrypted payload of a packet is not modified in any way during transit.

The SKe shared by the source and destination nodes is used to encrypt the payload of a SUPERMAN packet. This ensures that not only is the packet provided confidentiality against outsiders, but it is also indecipherable to nodes in the network. This provides secure communication within the SUPERMAN MANET. By allowing nodes to communicate privately in this manner, key reuse is minimised and nodes may be identified by the use of a specific key. If a key is used that a destination node may decipher, logically the packet must have been encrypted by the only other holder of that key. In this manner, it is possible to authenticate the source node by logical deduction, as the key itself provides a means of identifying a source node.

The same services are extended to broadcast communication in a one to many manner, by using the SKbe and SKbp keys held by each node to provide confidentiality, integrity and authentication to such transmissions.

Figure 6-11 visualises the outcome of applying confidentiality. The only role C can play in the communication between A and B is to discard the packet or forward it. Modification of the contents will result in B dropping the packet as it will not pass authentication unless A's SKe is used to encrypt the modified contents. C dropping incoming packets will only impact network performance if A is out of range of B, highlighting the need for point-to-point integrity checking to ensure that the route is legitimate.

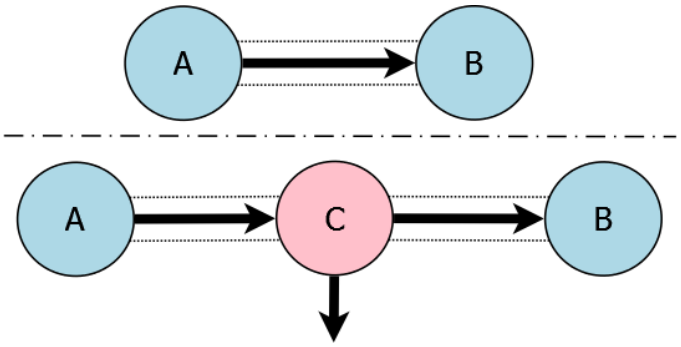

Figure 6-11 Diagram showing the effect of confidentiality and authentication on the man-in-the-middle attack

It should be noted that as SUPERMAN closes the network against outside observation, node C would not be able to participate in routing a message to B, as it does not have access to route information. This is because it would not have been allowed to participate in route generation, and cannot decipher such information easily or in a timely manner. This prevents abuse of the routing mechanism and stops malicious nodes from forwarding packets in an apparently legitimate manner to attempt to derive network information or inject modified data into the network.

6.4.5.2 Point-to-point Integrity and Authentication

End-to-end security only provides security services at source and destination, providing no information on if it has been relayed by intermediate nodes. It is possible that a packet

may be routed inefficiently if malicious nodes are able to replay valid packets to nodes that have not yet received them, causing problems with potential data collision and multiple instances of the same packet circulating in an attempt to reach destination.

Closing the network against outside access prevents malicious nodes from participating in all network and control services. This includes DTA, routing and any other form of communication that occurs within the VCN. However, to fully close the network confidentiality and integrity must be applied to all packets, and packets must be authenticated (traceable to a source or the last hop).

Sub-section 6.4.4.1 has detailed the means by which confidentiality is provided to packets in the VCN. Integrity, however, must be provided point-to-point, to provide a chain of custody along the route between a source and destination node. By digitally signing packets, nodes can provide evidence of origin. This provides proof of the last node to relay the message, allowing the identity of the previous node to be authenticated, as well as providing a means of checking that the contents of the packet have not been modified.

HMAC provides an encrypted digest (or tag) of the packet it represents that can provide both integrity and authentication services. When a node sends a packet to another, it digitally signs the encrypted payload, using the SKp shared between itself and the next node in the route. This ties the integrity of the packet to the identity of the transmitting node, allowing the receiver to authenticate the origin of the packet, and then ensure that the contents remain unchanged. This is performed without having to decrypt the payload, maintaining confidentiality between end points.

When a node receives a packet, it will strip the HMAC tag, and decrypt it using the SKp it shares with the sender. A hash of the encrypted payload will then be generated, and compared against the output of the decrypted HMAC tag. If these values match, the sender is authentic and the payload has not been modified. If the values do not match, the sender is either not a legitimate node, or the contents have been modified, and the packet will be discarded. This ensures point-to-point integrity, extending authentication services to the route. The result of this process is a chain of custody between source and destination, along the route, ensuring that only legitimate nodes have relayed the message towards its destination.

6.4.5.3 Securing Broadcast Messages

Broadcast messages are secured in a similar way to unicast messages, but use the SKb keys of the network, held locally by each member node. Many network services, including MANET routing protocols, require broadcast communication as a basis for topology generation and maintenance. Without a means of securely communicating these vital network services in an expedient and appropriate manner, the network is incapable of securely providing such services.

This means that nodes that have not yet joined the network cannot participate in secure routing. The authentication of a node with the network will result in the authenticating node sharing an SKb key, required to generate the appropriate SKbe and SKbp keys for the network via KDF. This will allow the node to participate in secure broadcast, allowing it to securely transmit routing information and other data that is dependent on broadcast communication. Confidentiality, integrity and authentication are provided, in the same way as defined in Sub-section 6.4.4.1 and 6.4.4.2.

When broadcasting, a node will use a BP, indicating that the contents of the SUPERMAN packet are intended for broadcast and that SKb derived keys should be used for integrity checks and decryption. When relaying messages, receiving nodes will use the same SKb derived keys to re-encrypt and sign the packet.

End-to-end security is provided by an AEAD construct using the SKbe key. Point-to-point security, in the case of messages that require network flooding or other multi-hop forms of broadcast, is provided by the SKbp key used to append a HMAC tag to the end of every broadcast packet. This tag does not change, as explained in Sub-section 6.4.4.2 when relayed by another node, as the SKbp is common to all nodes in the network.

6.4.5.4 Secure Tunnel

The result of point-to-point integrity and authentication, with end-to-end confidentiality and authentication, is a secure tunnel between the source and destination node(s). This includes integrity checking and authentication at every intermediate node for the length

of the route, ensuring that any misbehaviour on an intermediate node is detected and the propagation of malicious traffic is immediately stopped.

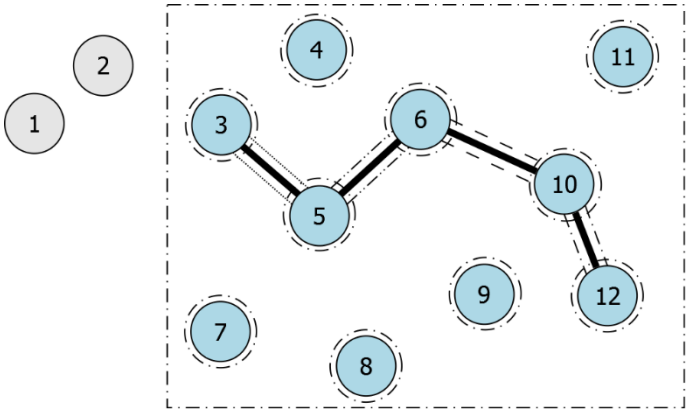

Figure 6-12 Diagram showing a secure tunnel between two nodes over three intermediate nodes with end-to-end and point-to-point security in a SUPERMAN VCN

Figure 6-12 demonstrates the application of end-to-end authenticated confidentiality and point-to-point authenticated integrity. Node 3 sends a message to node 12 via nodes 5, 6, and 10.

The solid line represents the encrypted transmission, which remains constant throughout the route. As previously defined, end-to-end authenticated confidentiality is not modified in any way by intermediate nodes, using the SKe shared by 3 and 12 to encrypt the payload and provide source authentication for the message.

The broken lines, which differ between each hop, represent the point-to-point authenticated integrity service of the framework. At each hop, the current and next nodes in the route communicate. The current node generates an HMAC tag using the SKp it shares with the next hop. This allows the sender of the packet to be identified by the key used and the packet to be integrity checked while remaining encrypted. This is repeated at every hop using a different SKp at each hop, until the destination is reached. If the source and destination are neighbours, the source and destination SKp is used to generate the HMAC tag.

6.4.5.5 Hierarchical Multi-MANET Networks

The VCN environment provided by SUPERMAN is based on the foundation of trust provided by a certificate. The certificates held by each node in the network are signed by a Trusted Authority (TA) that binds the network to a common trusted identity. However, it is possible that multiple networks may be friendly to each other, but not directly trusted. Cooperation is possible between such networks, through hierarchical certification.

A use case for such a system of certification is an emergency services network, making use of autonomous mobile nodes in some capacity. Fire service, police force and ambulance services may make use of independent networked systems, potentially with many nodes. However, it may be advantageous to share resources, or use another service's network as a means of extending your own networks range (communicating across intervening non-member nodes). Figure 6-13 shows an example structure of such a system.

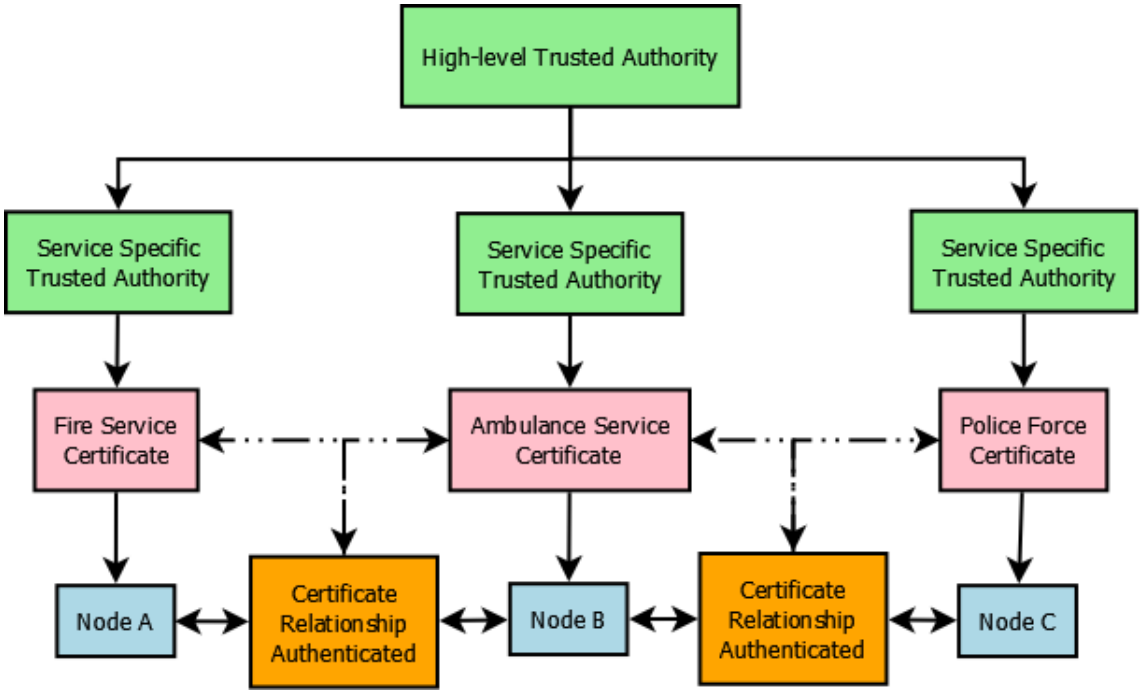

Figure 6-13 A hierarchical certificate authority scheme for emergency service autonomous MANETs

This is possible, in a SUPERMAN secured network, by having a single high-level authority that provides all lower-level certificates. By adopting a hierarchical certificate approach to VCN formation, it is possible to have multiple independent networks that may still trust each other due to the relationship between their different certificate's links to a higher authority.

In this example, each authority is related to a higher authority, representing a shared trust between the services each authority represents. Each authority signs certificates provided to it by nodes initialised into their respective networks, also providing hierarchical data relating to the relationship between the authority and all authorities above it.

Nodes wishing to communicate across other networks will exchange certificates and check their authenticity. If they are found to have been signed by a different trusted authority, the hierarchical data will be checked for parity. If both certificates share an immediate connection to a higher authority (that is one no more than one step upwards in the hierarchy), the nodes may trust one another and allow communication between and over their networks.

Figure 6-13 shows this by demonstrating three emergency services networks inter-communicating. This communication may be directed at the other networks, or routed over to reach another member of the same network that is more easily routed to over the intervening network (due to disconnect from its parent network or a more expedient route being available).

6.4.6 Section Summary

The SUPERMAN Framework for MANET security has been introduced, discussed and its security services explained. This is a novel security proposal, aimed at MANETs with the intent being to protect them against a multitude of attacks that exploit the open-medium used for wireless communication.

The core features of SUPERMAN are:

- The provision of a Virtual Closed Network (VCN).

    - A VCN is created by ensuring that all SUPERMAN nodes ensure that only other SUPERMAN nodes from the same TA (or set of TAs) are able to participate in the network. This involves all communication, anything communicated by a SUPERMAN node must be confidential until it reaches its destination, and integrity checked at every intermediate node. Only by ensuring that SUPERMAN nodes are fully compliant with the chosen security policy can the network be protected against passive attacks and potential intrusion based on derived information.
- End-to-end authenticated confidentiality.
    - Messages must remain confidential until they reach their destination. Destination nodes must be able to authenticate the source of a message to ensure that only legitimate SUPERMAN nodes are participating in network activity.
- Point-to-point authenticated integrity.
    - Every intermediate node must perform integrity checks on the HMAC tag of a SUPERMAN packet. This allows the last sender of a packet on a route to be authenticated by the next hop on the route to a destination node. By ensuring that each hop is authenticated allows the network to limit the propagation of potentially malicious packets by preventing the replacement of HMAC tags. Replay attacks are mitigated by using timestamps in conjunction with the HMAC tag to prevent trivial flooding of the network with replayed packets. Any change to the timestamp to avoid repeated values will be detected by an invalid HMAC tag. Unmodified timestamps will result in a repeated timestamp and the removal of the packet from circulation, reducing its impact on the network.
- Data security.
    - All data is secured in a SUPERMAN network. All data must comply with the previously discussed security policies and services to ensure that a VCN environment is maintained. By ensuring that no unencrypted communication is allowed, SUPERMAN prevents the snooping of network and service related data, locking out attackers.
- Expedient and robust access control.

- MANETs are often resource constrained. They have limited bandwidth and energy reserves, making low cost communication an important consideration. SUPERMAN enables all authenticated member nodes to perform node authentication and access control services. They may issue challenges to initiate certificate exchange or pass on credentials to nodes requesting certificate exchange on behalf of distant nodes. By reducing the number of retransmissions required to reach a distant node for security association, the total amount of communication and time spent communicating may be reduced.

Through these attributes, SUPERMAN may provide security services vital to the protection of MANETs against attack, by providing a closed network environment despite the use of an open medium. No specific hardware is required, routing protocols of any kind may be used (so long as they have routing tables) and security is considered to be applied in a homogenous manner across the network, with all privileged and services being applied equally to SUPERMAN nodes. In this way, attackers are denied trivial observation of the content of packets and are prevented from trivially assuming the identity of legitimate nodes to abuse the network.

6.5 Chapter Summary

The SUPERMAN framework has been proposed. Terminology, security services and the objectives of the proposed protocol have been discussed to provide a full account of the proposed framework and the rationale for its inception.

The relationship between node-level security and the privacy of the network has been a key point throughout this chapter. Individual nodes represent the smallest elements of the network and the only communication infrastructure available to the network. Nodes are required to initiate, relay and receive communication. By identifying that nodes are ultimately the only element of the network upon which a security framework may be deployed, the proposal of a virtual closed network (VCN) was able to be identified.

Virtual closed networks use an open medium for communication, but provide sufficient confidentiality, integrity and authentication services to legitimate nodes to prevent trivial observation of communication or insertion of malicious nodes into the network. Security must be applied homogenously across the network. Every node must adhere to a common set of security policies, enforcing the service extended by the security framework in a uniform manner so as to avoid the need for synchronisation between nodes for session specific security. By ensuring all nodes cooperate with globally defined security policies, the network may be virtually closed by preventing unencrypted communication and ignoring any packets that cannot be authenticated.

By allowing all nodes recognised as legitimate to operate as gate keepers for entry to the secure network, SUPERMAN may reduce the cost of security by preventing the formation of lengthy routes for security associations between nodes. By allowing nodes to join via any authenticated member of the SUPERMAN network, hop count in the initial certificate exchange phase is also reduced to a single hop. This may limit the impact of complex challenge-response style communication on the wider network.

Having proposed the SUPERMAN framework, the next chapter will focus on the definition of a test plan, experiments, results of those experiments, and empirical analysis of the SUPERMAN framework.

7 TESTING & RESULTS: SUPERMAN

7.1 Chapter Introduction

Chapter 6 proposed SUPERMAN, a novel security framework targeted at MANETs. The aim of the proposed framework is to provide security in highly dynamic mobile ad-hoc networks, while using as little in the way of network resources as possible. It has been identified that security requires additional resources to secure packets. Security services may also require a set up phase, in which nodes join the network securely and form security associations with other members of the network. It has also been identified that the primary issue with MANETs is their use of an open medium for communication; leaving them vulnerable to passive observation and attacks that are based on derived network information.

The SUPERMAN framework has been proposed as a means of providing a virtual closed network (VCN). By closing the network at the node level, ensuring that all nodes in a SUPERMAN network follow globally standardised security policies, the open medium problem may be mitigated. If confidentiality services are extended to all data, observation of said data becomes non-trivial. Extending integrity services to all packets mitigates the effects of packet flooding and DoS attacks.

This chapter focuses on proposing a series of experiments, mathematically modelling and simulating SUPERMAN and comparable security approaches. Through the undertaking of these experiments, and the analysis of results obtained from them, quantitative data may be derived regarding the primary characteristics of SUPERMAN. These include effects on packet size, and communication events required for DTA and throughput. Qualitative results regarding the security services provided by SUPERMAN compared to comparable approaches to MANET security are provided through analysis of the quantitative results.

7.1.1 Chapter Layout

This chapter is presented as follows:

- Section 7.2 outlines the experimental methodology for the modelling and simulation of SUPERMAN, IPsec, SAODV and SOLSR for the purpose of allowing comparative analysis of quantitative data.
- Section 7.3 provides results from the simulation of SUPERMAN, IPsec, SAODV and SOLSR.
- Section 7.4 provides a comparative analysis of the findings presented in Section 7.3.
- Section 7.5 summarises the chapter.

7.2 Experimental Methodology

Section 6.3 maps the hypotheses developed in Section 3.4 to the core principles of the SUPERMAN security framework. These hypotheses provided the foundation upon which the theoretical framework was based, and now provide the means by which testable elements and results can be identified.

Each node may be seen as an end-point, router and security provider, which may be protected only by securing each and every node, instead of relying on infrastructural approaches to security (such as router firewalls).

The above excerpt is a short-form version of the first hypothesis relevant to the inception, proposal and testing of SUPERMAN. It concerns the closure of the network to unknown nodes, unless they are able to successfully pass through access control and authentication routines to prove that they are legitimate members of the network. Due to the highly distributed nature of MANETs, infrastructural approaches such as using specialised ‘secure’ nodes to gate entry to the network are impractical, due to the variable position and thus availability of such specialised nodes.

By providing end-to-end confidentiality services and point-to-point integrity services between all nodes, for unicast, multicast and broadcast communication, the network may be closed against outside observation and access control services provided to control node entry to the network

The second hypothesis considered in the context of the SUPERMAN framework is that shown above. In addition to providing access control and authentication services, vital security services must be extended to end-to-end communication. This includes broadcast, multi-cast and unicast communication, providing confidentiality and integrity services for all legitimate nodes.

Combined with access control enforced on every node, confidentiality and integrity services must provide an environment in which members of the network may communicate without intercepted messages being a viable vector for attack. All data must be encrypted and integrity checked to ensure that only authorised members may expediently act on critical information.

The testable elements related to these hypotheses have been identified as:

- Number of nodes.
- Comparative analysis of SUPERMAN and IPsec.

The number of nodes in a network is directly related to the complexity of the topology, as the routes in large dispersed networks of drones are likely to be longer than those of dense or small networks. By increasing or decreasing the number of nodes participating in security processes, the effects of network size and communication complexity can be observed.

Comparative analysis of SUPERMAN against IPsec will be performed to determine the relative security costs associated with the two approaches. Communication events and measurements of the number of bytes passed out of the network layer will allow for the analysis of whether SUPERMAN can deliver similar security services, such as a closed network environment and end-to-end confidentiality, for a similar or lower amount of data passed out of the network layer. This is an important element to observe, as the ability to provide adequate security at a cost that matches the capabilities of the target network is a key consideration when discussing MANET security. The rationale for only

observing the bytes leaving the network layer is that the data link and physical layer are user defined and can vary greatly, while SUPERMAN is expected to operate on most configurations of the OSI model without dependencies on higher of lower layers using a given protocol.

The resulting variables, which will be analysed to determine the comparative characteristics of SUPERMAN compared against IPsec, have been identified as:

- Number of communication events.
- Number of bytes transmitted.

These three variables have been identified as indicators of cost. The cost of security can be broken down as follows; the number of transmissions required to provide a service to the whole network, the amount of data required to provide that service, and the delay incurred on a given network interface by the provision of that service. Although security is a vital consideration and a core service in many networks, it does use resources that might have been allocated elsewhere, DTA for example. Therefore justifying the cost of security first requires that the cost is identified, and then put into context. By analysing the above variables that will result from the modelling and simulation of SUPERMAN, this question can be answered. By comparing SUPERMAN with IPsec (and secure routing protocols where appropriate), the relative cost of different approaches to security can be observed and conclusions drawn from the resulting analyses.

IPsec has been selected for comparison as a security framework that provides a suite of security services, including integrity, confidentiality and authentication, to end-to-end communication. MANET focused frameworks focusing on secure routing and reputation-based systems exist, but do not represent the same ‘full-security’ philosophy that SUPERMAN adheres to. Therefore IPsec is seen as a representation of the cost of providing security to all communication between nodes in a network, by securing the links between nodes, instead of focusing on one element of the network exclusively (such as routing or control communication).

The following experiments will be performed; modelling of network characteristics, and simulation of SUPERMAN and IPsec for the purpose of comparative analysis. These

experiments will identify the attributes and comparative traits of SUPERMAN, when compared against an established security framework.

7.2.1 Simulation of SUPERMAN

Simulation of the SUPERMAN framework will be undertaken to investigate three key areas:

- The number of communication events required to securely join the network (SUPERMAN).
- The number of communication events required to for security associations between all nodes.
- The total number of communication events required to perform Internet Key Exchange (IKE) and Security Association (IPsec), and network join and Security Association (SUPERMAN).
- The number of bytes transmitted whilst exchanging certificates to join the network (SUPERMAN).
- The number of bytes transmitted to secure communication between all nodes.
- The total number of bytes required to secure the network, including IKE and Security Association (IPsec) and network join and Security Association (SUPERMAN).

These areas of investigation will be analysed by comparing the results of SUPERMAN protected communication against that of IPsec. Where a phase of securing the network is not directly equivalent (for example SUPERMAN's network joining process and IPsec's IKE phase) the total cost of security is analysed, with the phases included in that total instead of being directly compared. This is to avoid the comparison of non-equivalent functionality in the security frameworks being compared, while still allowing a total cost of security to be determined.

The number of communication events required by either framework is referred to as the cost of communication throughout this chapter. As discussed in Chapter 5, the time and network resources spent sending each message between nodes can be described as communication events, or individual transmissions assumed to be within the MTU limit of the network interface. By tracking the number of communication events required to provide network and security services, the communication cost of those services may be identified and analysed.

7.2.1.1 Simulation Parameters and Rationale

Table 7-1 outlines the parameters of the simulation. All simulation will be performed using MATLAB. The nodes are considered to be static for the purpose of this simulation. It is assumed that all packets arrive intact and without bit-error or loss.

Table 7-1 SUPERMAN Simulation Parameters

Number of Nodes:	10 - 100
Routing Algorithm:	Dijkstra (shortest path)
Number of Iterations:	100
Simulation Area:	100m x 100m
Communication Range:	50m
Max Hop Count:	5
Random Seed integer:	11
Pseudo-random Number Generation Algorithm:	Mersenne Twister
Key Share Size (Bytes):	128 and 256
Certificate Size (Bytes):	1013 and 1275
SUPERMAN header size (Bytes):	5
IPsec authenticated header (AH) size (Bytes):	8

The number of nodes is set to a range of 10 to 100. This allows for the observation of the effects of increasingly large networks on the complexity of communication, and therefore the effects of complexity on the ability of the security approaches being investigated.

To allow for the shortest routes between nodes to be calculated, Dijkstra's algorithm has been incorporated into the network generation and topology definition procedures used to create the simulation environment. This algorithm uses the connectivity graph of the

network to allow the topology of the network to be defined, and to ensure that all nodes are connected via no more intermediate nodes than allowed by the maximum hop count. Networks not meeting this requirement are regenerated, to ensure that only networks with viable routes over which to communicate are simulated. SUPERMAN does not require that a specific routing protocol is used, which is why all routing is pre-calculated using Dijkstra's algorithm instead of being included as part of the simulation.

Each experiment is run 100 times for each network. This ensures that irregularities brought about by the random generation of the network do not characterise the general trend of the communication performed during the simulation. By conducting experiments in 100 iteration batches, a high degree of confidence can be attained by providing an averaged value for those iterations, while still allowing analysis of anomalous data should a deviation from the trend remain after that many iterations. Iterations are randomised as to where nodes are placed prior to topology definition, using Mersenne Twister (Matsumoto & Nishimura 1998) initialised with a seed of value 11. This particular PRNG has been chosen due to its high usage in a variety of applications, passing many tests for statistical randomness, including the Diehard tests (Alani 2010). This will provide a reliably random distribution of nodes throughout the simulation environment for each instance of simulation.

A simulation area of 100 by 100 metres has been created for these simulations. Nodes are given a 50 metre communication range to ensure that there are multi-hop routes throughout the mission area. The combination of surface area and communication range generates a wide range of diverse topologies over the iterations run for each experiment, again allowing for analysis of general trends in highly changeable networks.

Diffie-Hellman key shares are simulated with lengths of 128 bytes or 256 bytes, representing a baseline standard of security and a high security variant. Network key shares, used for broadcast communication in the MANET are equal to Diffie-Hellman key shares in length. Certificates are simulated in sizes equivalent to the output of an open SSL (X.509 standard) certificate generation process, with key shares set to the defined values. This results in two different certificate sizes. The total cost of security (in bytes) will be identified, allowing the network resource requirements of each security bracket to be identified.

The addition of security to data packets will cause them to increase in size to accommodate the SUPERMAN header and footer required by the SUPERMAN framework. IPsec has similar requirements, notably the need for authenticated headers and footers in AH modes of operation. In ESP mode, HMAC tags are still required for integrity and source authentication.

7.2.1.2 Secure Routing

Secure routing protocols exist to facilitate the trustworthy, reliable and safe generation of routes between nodes in a MANET. These protocols do not provide protection to data transmitted over the route, but they do provide guarantees of legitimacy for nodes on that route. Such security measures are considered a requirement for MANETs operating in all but the most trusted environments, to prevent malicious manipulation of topology generation mechanisms.

IPsec does not provide any facility for routing security, as it does not account for the route taken by data, only the end-points involved. SUPERMAN, however, does provide protection to routing packets, by providing secure broadcast communication within the VCN. Therefore, SUPERMAN can be compared with secure routing protocols to determine the cost of providing secure routing services under a given protocol.

Two routing protocols, AODV and OLSR, have been selected to represent reactive and proactive routing respectively. These protocols have been selected as they are popular MANET routing protocols for MANET research, both listed by the IEEE as standardised MANET protocols in their respective categories (reactive and proactive).

Their secure implementations, SAODV and SOLSR, are compared with SUPERMAN secured routing packets using the same base protocol. The insecure routing protocols (AODV and OLSR) are also simulated, to allow for the calculation of a cost of security variable, based on the number of additional bytes required to provide security services.

The simulations follow the parameters outlined in Table 7-2, reiterated below:

- 10-100 node networks are simulated.

- AODV, SAODV and SUPERAODV are simulated to determine the cost of security under a specific reactive routing protocol.
- OLSR, SOLSR and SUPEROLSR are simulated to determine the cost of security under a specific proactive routing protocol.
- Nodes are randomly placed in a 100 by 100 metre simulated area.
- All simulations run for 100 iterations.

7.2.1.3 Secure Distributed Task Allocation

A comparison of SUPERMAN and IPsec in the context of Distributed Task Allocation (DTA) will be undertaken to analyse the comparative cost of security when protecting control services that facilitate network autonomy. MATLAB will be used to simulate DTA to determine the size of additional communication overhead (in bytes) over two scenarios:

- CBBA task allocation involving 18 nodes.
- CF-CBBA task allocation involving 6 clusters of 3 nodes (18 total).

Networks of 18 nodes have been chosen to provide an insight into how multi-hop communication across the network affects DTA. Results of preliminary simulations showed that 18 node networks formed multiple multi-hop connections between nodes, providing a small network capable of solving problems in a timely manner while allowing communication cost to be analysed in light of routes of variable length. Both DTA processes will have a task list of between 1 and 50 tasks, all of which must be assigned for the DTA process to be considered complete. This will allow for further analysis of security overheads in increasingly complex problem domains.

7.2.1.4 Mathematical Fundamentals of the Simulation

Two equations are used as the basis for modelling and simulating the number of bytes used to secure network and control service transmissions. Equation 7-1 shows the means

by which *x*, the total additional bytes required by a given security protocol, may be calculated. The function of *c* represents the number of rounds required by a given consensus based distributed task allocation algorithm. The number of nodes is represented by *n*. The header and signature requirements of the framework in question are represented by *h* and *s* respectively. The probability of a packet being delivered is represented by the variable *p*. In the case of the experiments undertaken in this thesis, *p* is equal to 1, as it is assumed that the network operates on a perfect channel. This equation holds true for any non-clustered method of distributing tasks throughout a MANET.

$$\boldsymbol{x = \frac{\Big(f(c).\big(n(n-1)\big)\Big).(h+s)}{p}} \qquad \textbf{(Equation 7-1)}$$

This equation is used to model the amount of data required to ensure the reception of a message between two nodes directly or over a number of intermediate nodes. It is used in the simulation to provide a mathematical basis for the simulation of data transfer between nodes as whole packets. When calculating the amount of data required to complete a DTA process, the sum of all communications represents the total number of bytes required by the network in question. Each instance of communication is represented by the equation above, with the result of all such calculations providing the total bytes required to achieve consensus.

Equation 7-2 represents the same operation, but for clustered task allocation algorithms. The total number of bytes, *y*, is the sum of all cluster allocation transmissions (represented as instances of *x*). The variable *p* of *x* represents the cluster head allocation of CF-CBBA, which is performed prior to pushing the resulting task lists to the cluster level for final allocation among cluster members.

$$\boldsymbol{y = \left(\sum_{1 \le i \le L} x(i)\right) + x(p)} \qquad \textbf{(Equation 7-2)}$$

Both of these equations highlight that the primary contributing factor to the additional overhead incurred by data packets using SUPERMAN or IPsec, is the size of the header and signature. A symmetric block cipher is used in these simulations, to avoid the padding that asymmetric block ciphers require. This saves on total packet size in packets which are not exactly divisible by the block size.

The number of additional bytes required to provide the headers, footers and encryption for routing security is used to define the security overhead of a given approach in this context.

7.2.2 Section Summary

Simulation will be undertaken to provide results showing the number of communication events and the amount of additional data required by SUPERMAN and IPsec to perform critical network services which allow for the provision of a virtual closed network environment.

Additional simulation will be performed to allow analysis of the data requirements of control and network services when considering the cost of adding security. CBBA and CF-CBBA will be simulated to demonstrate the cost of IPsec and SUPERMAN security services for each DTA algorithm. SAODV and SOLSR will be used to provide a basis for comparative analysis against SUPERMAN secured AODV and OLSR routing processes.

Analysis of these results will follow, highlighting key observations regarding the cost of security in terms of additional control packets and the additional packet size required to add security to existing data packets in the network.

7.3 Results of Simulation

7.3.1 Communication Events: SUPERMAN and IPsec

Figure 7-1 shows the results of simulating the network initialisation phase of SUPERMAN. This phase involves authenticating nodes through certificate exchange in a one-time process that results in nodes forming or joining a network, including broadcast keys and security association between the two nodes engaged in certificate exchange.

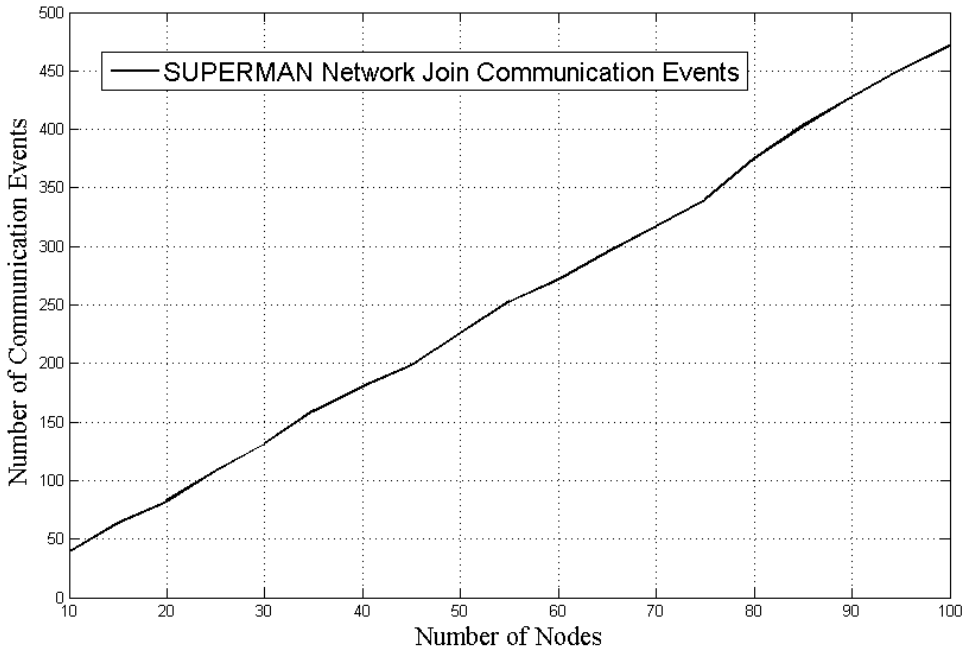

Figure 7-1 Graph showing the number of communication events required by SUPERMAN to allow all nodes in a network the join the VCN

As network initialisation under SUPERMAN is a one-time process, the number of communication events required to join the network is low, with 100 node networks only requiring 471 communication events for all nodes to have joined the network. At this point, nodes are not yet securely associated with other nodes, except for with the neighbouring nodes that engaged in certificate exchange. At this point, SUPERMAN nodes are able to participate in routing and broadcast communication, but need to securely

associate with other nodes to fully close the network. The initialisation phase is the primary means by which access control is managed.

Figure 7-2 shows the communication events required to complete the Internet Key Exchange (IKE) phase of IPsec. IPsec differs greatly from SUPERMAN in that it does not secure a network, but instead secures connections between nodes. The IKE phase is required to share authentication credentials (such as certificates) to provide data needed to begin secure sessions between end-points.

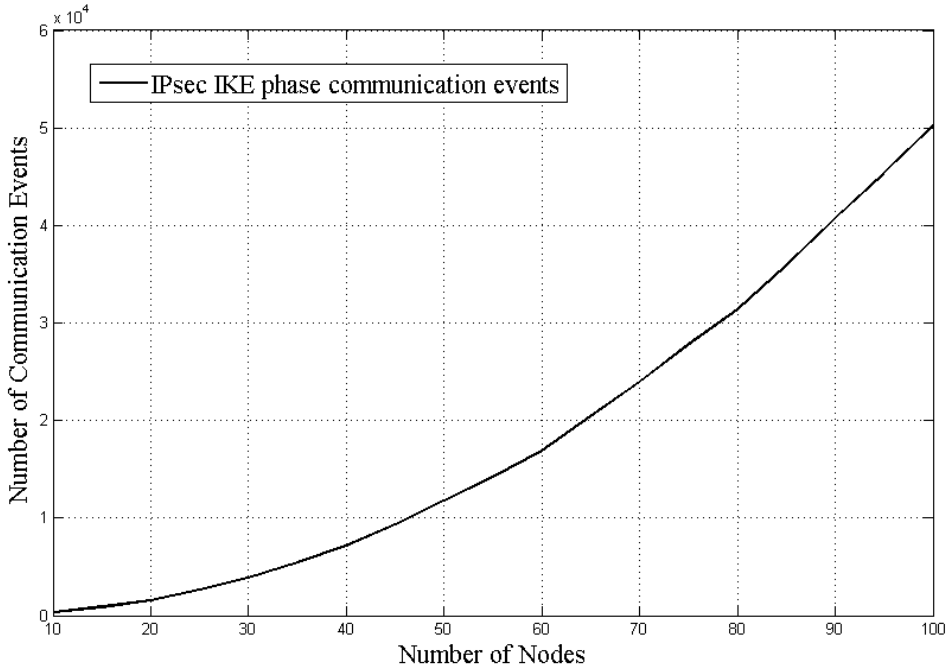

Figure 7-2 Graph showing the number of communication events required to complete IKE between all nodes in a network

The IKE phase involves sharing credentials between all end-points that need to securely associate in the future. In this case, the IKE phase is carried out between all members of the MANET being simulated. This can generate a substantial amount of communication events, with an average of 50,023 events being recorded for 100 nodes. This is substantially more than is required for SUPERMAN VCN initialisation, but it must be stressed that the two processes are not equivalent. SUPERMAN generates a VCN environment for all initialised nodes, IPsec exchanges security credentials for future use on an end-to-end basis.

Figure 7-3 provides the first comparison of SUPERMAN and IPsec, looking at the results of security association simulation. Security association involves the exchange of key shares or other public cryptographic data (henceforth referred to as security credentials) to generate a secure link between two nodes. Both IPsec and SUPERMAN require this exchange of information to facilitate secure communication between end-points.

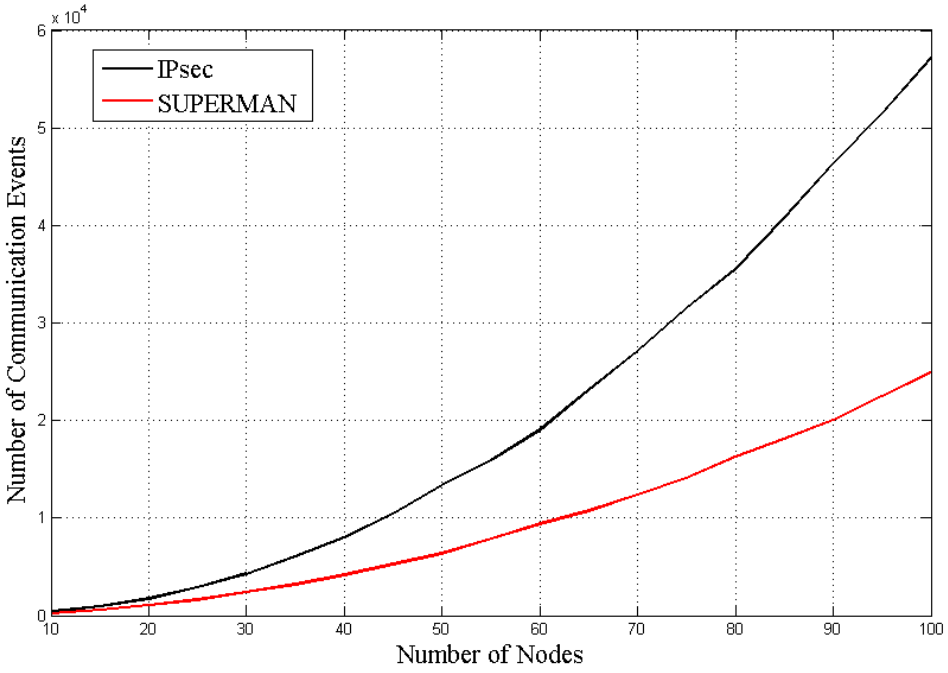

Figure 7-3 Graph showing the number of communication events required by IPsec and SUPERMAN to form security associations between all nodes in networks of various sizes

A significant difference between IPsec and SUPERMAN can be observed immediately. Although both approaches aim to secure the links between nodes regardless of the number of hops between nodes, SUPERMAN possesses a referral mechanism, allowing nodes possessing the security credentials of the destination node towards which they are propagating a message, to reply on its behalf if they have previously performed security association with the origin and destination. As more end-points form secure links, this mechanism becomes more noticeable, effectively reducing the length of routes when requesting security credentials over multi-hop routes.

For a 100 node network, SUPERMAN requires an average of 24,120 communication events to securely associate all node. IPsec requires 57,803 events to securely associate all nodes in a 100 node network.

Figure 7-4 shows the total cost of security, in terms of communication events, for both IPsec and SUPERMAN.

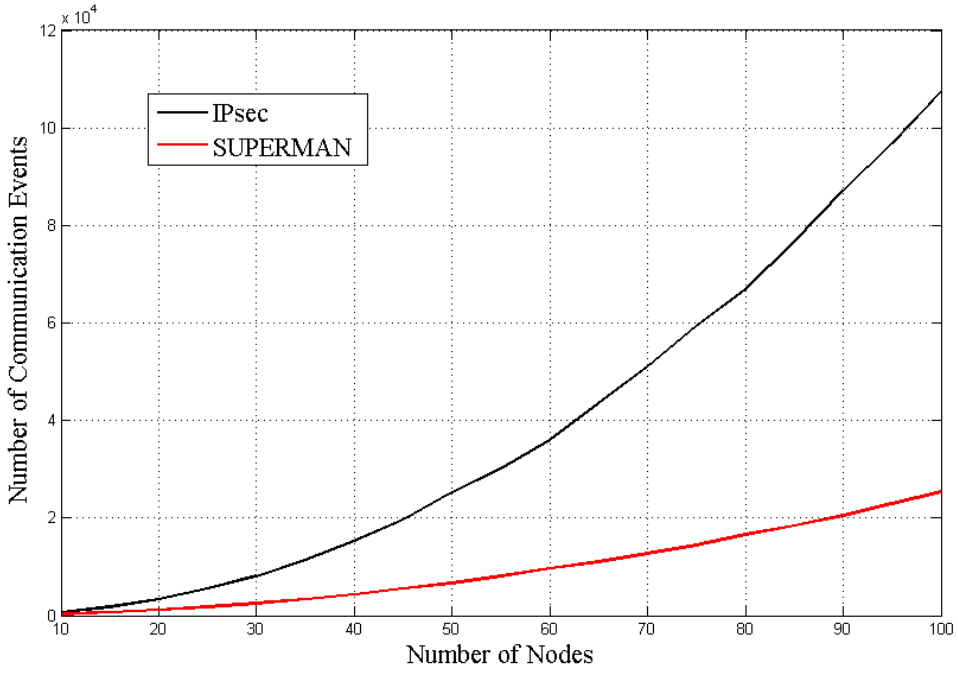

Figure 7-4 Graph showing the total communication events required to provide a fully secured network environment, under IPsec and SUPERMAN

Though it has been previously pointed out that the initialisation phase of SUPERMAN and the IKE phase of IPsec cannot be considered equivalent (and thus are incomparable), the total cost of security may still be compared. The total cost of security, in this case, is the combined total of communication events required to drive initialisation, IKE and security association, providing a fully secured network under SUPERMAN or IPsec.

IPsec does not scale well, when compared to SUPERMAN, which is logical considering its primary use as a means of securing end-to-end communication. When used to secure a network fully, significant overhead is generated in terms of control packets required to set up security features. An averaged total of 100,789 communication events is observed

for networks of 100 nodes, and a rapid increase in the number of required events can be seen in networks of 30 nodes or more.

SUPERMAN, due to its referral mechanism and one-time (per joining node) initialisation phase, generates significantly less communication events and shows much better scalability. An average total of 24,612 communication events can be observed for networks of 100 nodes. This total is consistently below that of IPsec, with SUPERMAN requiring only 24.3% of the communication events needed by IPsec for 100 node networks.

7.3.2 Total Bytes Transmitted: SUPERMAN and IPsec

Figure 7-5 shows the number of bytes transmitted when performing network initialisation under SUPERMAN. Two key-share sizes are simulated, representing two different levels of security. 1024-bit and 2048-bit key shares are used to simulate the initialisation phase, allowing observation of the effects of increased cryptographic complexity on the data requirements of the SUPERMAN framework.

As expected, 2048-bit keys require more data to be sent to complete network initialisation than 1024-bit keys. The former requires that 3.74 megabytes of data are sent to secure a 100 node network completely. The latter requires 2.67 megabytes. Initialisation with 1024-bit key shares requires 28.7% less data than initialisation with 2048-bit key-shares in a 100 node network.

It can be observed that the disparity between the key sizes will grow as the network increases in size. However, in smaller networks, the cost of increased security may not be as high, possibly indicating the feasibility of highly secured small networks, with smaller key shares being preferred for larger networks.

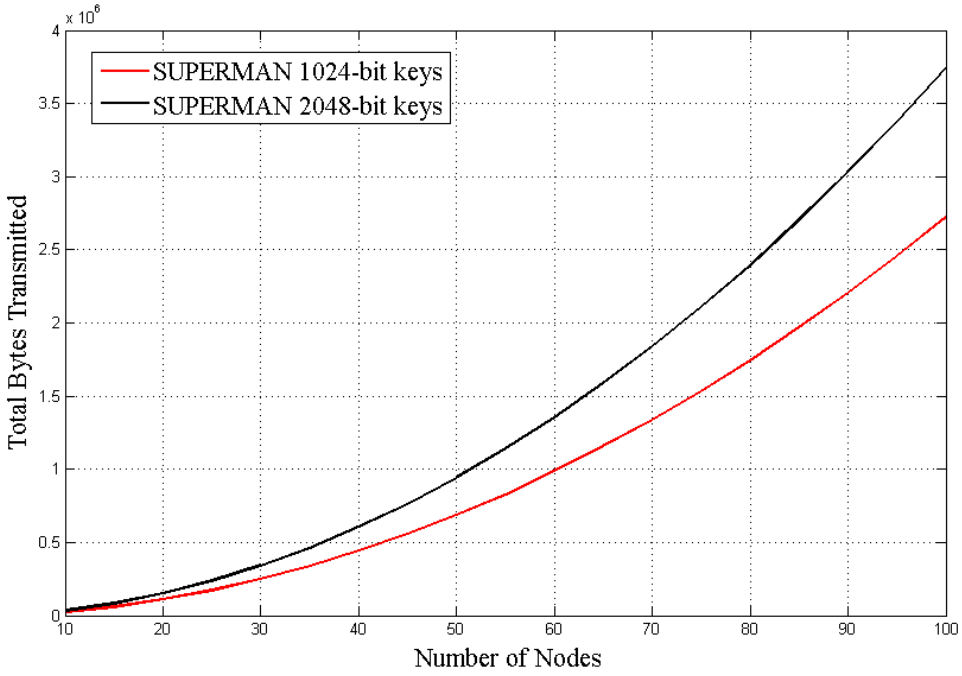

Figure 7-5 Graph showing the number of bytes transmitted during the initialisation phase of a SUPERMAN VCN

Sub-section 7.3.1 pointed out that the initialisation phase of SUPERMAN and the IKE phase of IPsec are not directly comparable. This remains true, but for the sake of completeness the simulation of IPsec IKE has been included to contextualise the total cost of security (in terms of data requirement) at the end of this sub-section).

Figure 7-6 shows the data required by IPsec to perform IKE on networks of various sizes. IKE is very data intensive on large networks, as it is intended to share security credentials between end-points, not provide membership in a network. As a result, every node must perform IKE with every other node.

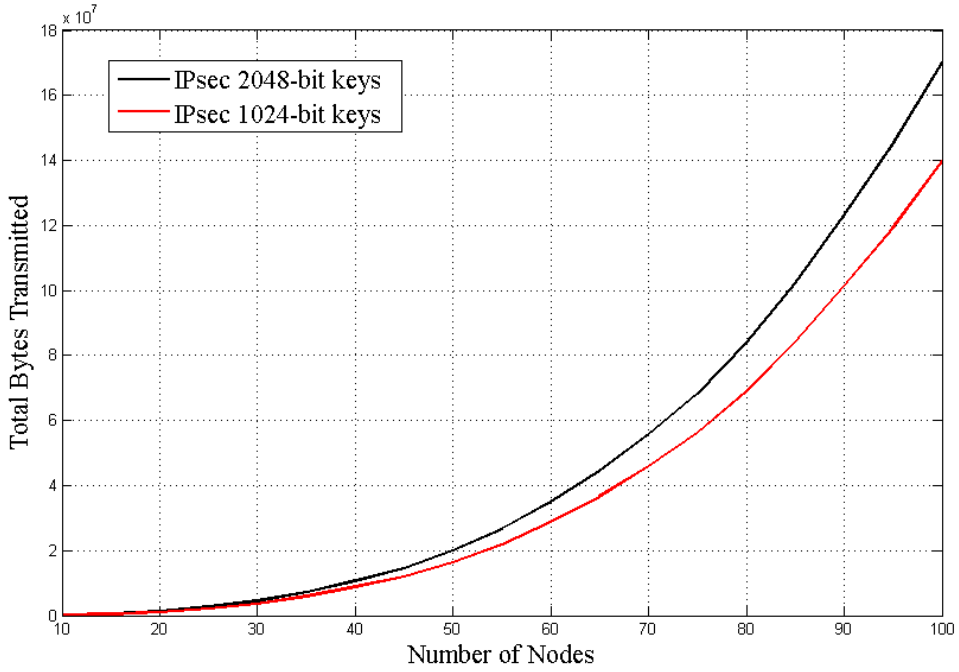

Figure 7-6 Graph showing the number of bytes transmitted during IKE under IPsec

The high cost of IKE is due to its use in communicating security credentials between end-points instead of authorising network access. IPsec IKE begins to accrue a significant amount of traffic as the network increases in size, as it is required to perform IKE between every communicating pair of nodes. If the network as a whole is to benefit from IPsec, this means that all nodes must perform IKE, sometimes over multiple hops.

Figure 7-7 shows the data requirements of SUPERMAN and IPsec for security association, using the previously established key-share sizes. All nodes must associate with each other for the network to be considered fully secured, though in a real-world application it is not required that the network be in such a state, so long as actively communicating end-points and intermediate nodes are associated.

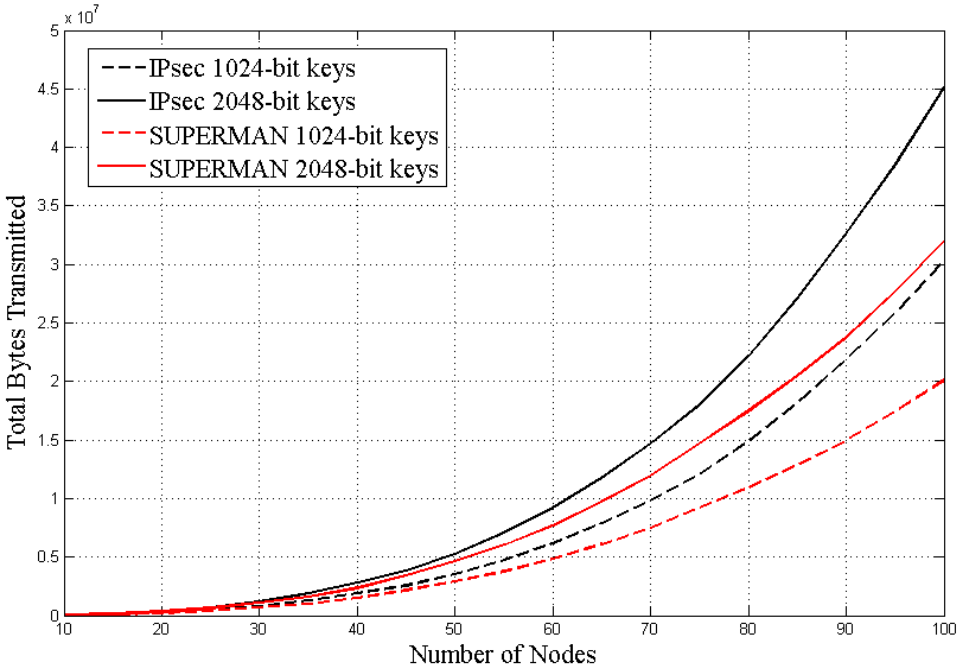

Figure 7-7 Graph showing the number of bytes transmitted by nodes forming security associations under IPsec and SUPERMAN

SUPERMAN is observed as being consistently less data intensive than IPsec in either configuration. SUPERMAN requires 32.3 megabytes to be sent during security association for 2048-bit key-shares when securely associating 100 nodes. For networks of the same size, SUPERMAN using 1024-bit key-shares requires an average of 20.3 megabytes.

IPsec, when securely associating 100 nodes, requires that 45 megabytes of data is transmitted when using 2048-bit key-shares. For 1024-bit key-shares, IPsec requires an average of 30.5 megabytes.

Figure 7-8 shows the total data cost of security, in bytes, for IPsec and SUPERMAN. Network initialisation and security association phases are accounted for under SUPERMAN. IKE and security association are included for IPsec. This provides a total cost of security for each approach, giving a figure representative of the data cost of either security framework when completely securing a network of between 10 and 100 nodes.

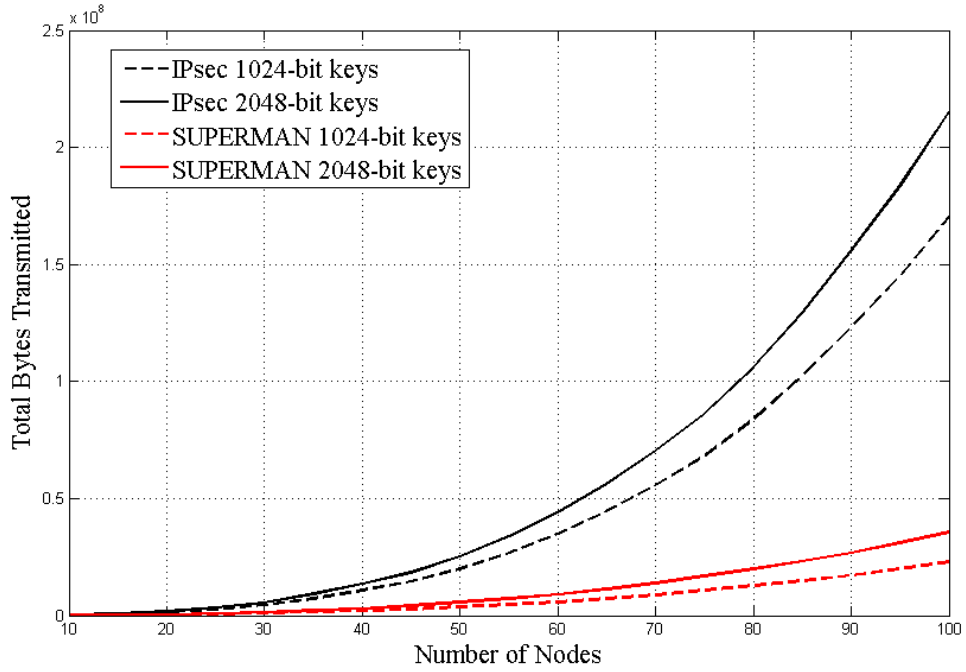

Figure 7-8 Graph showing the total bytes transmitted when fully securing a network under IPsec and SUPERMAN

The cumulative cost of IKE and security association make IPsec an expensive prospect for large networks. Networks of 100 nodes require that IPsec, using 2048-bit key-shares, generates and average 212 megabytes of data to fully secure a network. Using 1024-bit key-shares reduces this to 162 megabytes, still a significant cost.

SUPERMAN benefits from being designed for MANET security. One-time network joining, network merging and referred security association mechanisms all serve to reduce the effective length of routes and cut down on the requirement to repeatedly share larger security packets, such as those containing certificates and key-shares. In a 2048-bit key-share configuration, SUPERMAN requires 36 megabytes. That is 5.9 times less data than IPsec in a similar configuration. Using 1024-bit key-shares, SUPERMAN requires only 22.1 megabytes of data. In both cases, the MANET-focused network security mechanisms of SUPERMAN serve to significantly reduce the cost of security, while providing a completely closed VCN environment for all member nodes.

7.3.3 Secure Routing

Routing is a vital process for MANETs, generating, defining and maintaining the topology of the network and informing nodes of the routes available to any point in the network. Securing routes is very different to securely routing. The former requires that the route between nodes extends security services to the data passing through it. The latter requires that nodes advertising availability for a given route are trustworthy and authentic. In this sub-section, secure routing is the focus of the two simulations that have been undertaken.

SUPERMAN is able to secure routing packets, providing security services to the routing process. AODV and OLSR are MANET routing protocols representing reactive and proactive routing approaches respectively. SUPERMAN applied to each of these protocols is compared against secure implementations of each, SAODV and SOLSR. This provides a basis upon which the relative cost of securing routing using one approach or another can be analysed.

In each simulation, routes are generated between all nodes. It is understood that reactive protocols would normally generate routes on-demand, instead of generating the entire topology, but for the sake of ascertaining a whole-network cost of secure routing, it is assumed that all nodes must be able to route to each other for these simulations.

Figure 7-9 shows the simulation of a reactive routing protocol, AODV. To form routes between all nodes, the AODV protocol must broadcast route-request messages until the destination node receives one, at which point a route-reply is propagated back towards the source, constructing the route. This is not a simple means of generating a route, but does provide benefits in terms of reliability and the potential to form multiple different routes should trust-metrics or other variables make the first reported route undesirable.

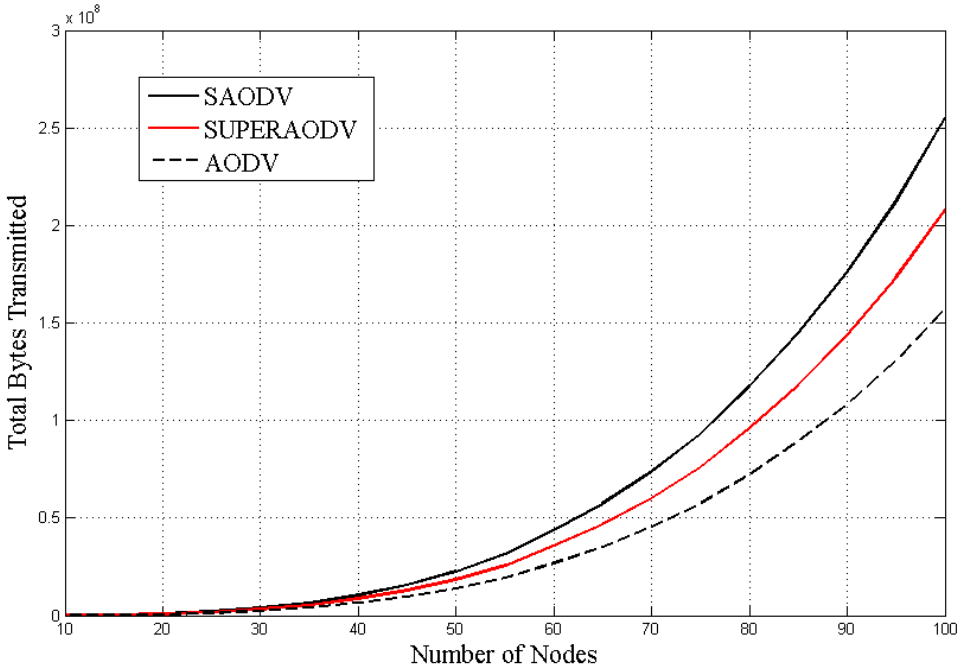

Figure 7-9 Graph showing the total bytes transmitted when forming routes between all nodes in a network using AODV, SAODV and SUPERAODV

AODV, without security, requires that an average of 150.7 megabytes of data are transmitted to form routes between 100 nodes. Though this may seem a large figure, this does include the route data accrued during the route-request process and the propagation of that information back towards source after the shortest route to destination is found. That figure also includes the route-requests that propagate throughout the network until they loop back on themselves (and are dropped). Therefore a significant portion of the figure stated above may be wasted packets sent during the request phase of route formation.

SAODV increases the size of routing packets by adding a digital signature and sequence numbering to the process. A hash is also required for the hop count, to protect the data from trivial manipulation by third parties during transit, and binding it to the identity of the transmitting node. As a result, SAODV requires an average of 250.1 megabytes to form routes between 100 nodes.

SUPERMAN does not employ the same security measures as SAODV. SAODV is assumed to not operate in a VCN environment, and therefore does not benefit from access

control and authentication features intrinsic to SUPERMAN. SUPERMAN encapsulates routing packets in a SUPERMAN broadcast packet (BP), extending point-to-point integrity, end-to-end confidentiality and source authentication (bound to the network broadcast keys). As a result, no additional modifications must be made to the routing packet, it may be added directly to the BP as a payload. This reduces the total size of the packet at the network layer, with SUPERMAN requiring an average of 206 megabytes to securely form routes between 100 nodes.

OLSR is a proactive routing protocol, which floods routing packets periodically, to update relay nodes responsible for routing messages between nodes. It is a lightweight protocol, due to the requirement that the topology is frequently regenerated, producing a lot of traffic over time, but with a very small profile in terms of packet size. Figure 7-10 shows the results of simulating the formation of routes between all nodes in various networks.

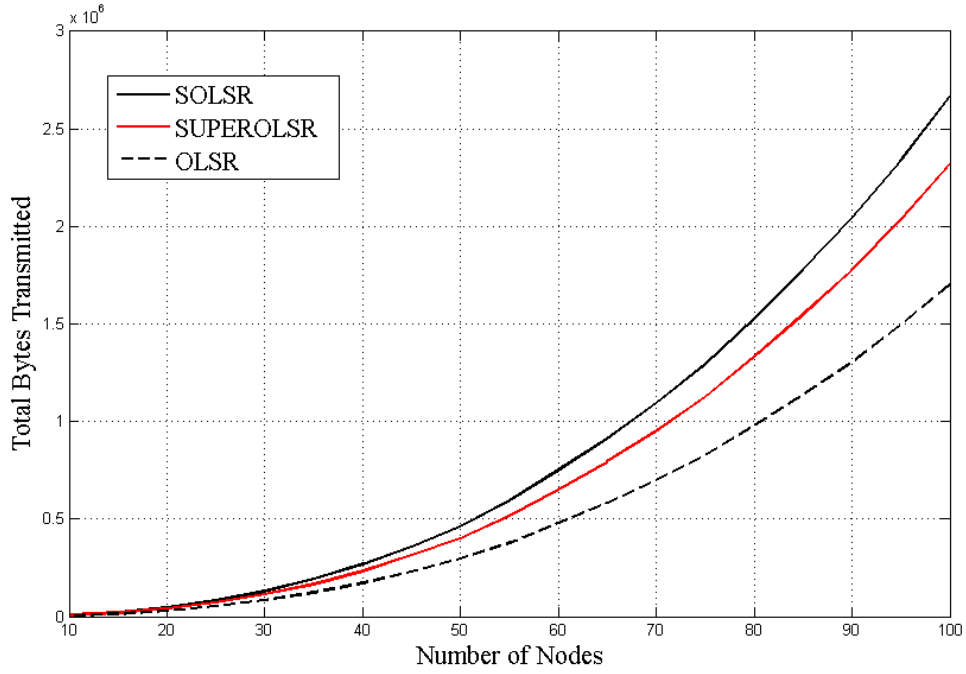

Figure 7-10 Graph showing the total bytes transmitted when forming routes between all nodes in a network using OLSR, SOLSR and SUPEROLSR

OLSR, when forming routes between 100 nodes, requires an average of 1.64 megabytes of data. This is a significant reduction when compared with AODV, due to the lightweight packets and simple algorithms used to generate OLSR routes.

SOLSR adds some more complex behaviour, as well as providing a timestamps, random challenge values and digital signatures. SOLSR requires that nodes reply to route requests, instead of passively updating as they do under OLSR. This increases the amount of communication required significantly. SOLSR requires an average of 2.63 megabytes of data to form routes between 100 nodes.

SUPEROLSR preserves the simple behaviour of OLSR, by encapsulating OLSR packets in SUPERMAN broadcast packets (BP). This provides source authentication, point-to-point integrity and end-to-end confidentiality. By preserving the simple behaviour of OLSR, SUPERMAN decreases the cost of security when compared to SOLSR, but the increased packet size increases the data requirement relative to baseline OLSR. SUPEROLSR requires 2.31 megabytes to securely form routes between 100 nodes.

7.3.4 Secure Distributed Task Allocation

To secure DTA, each bundle exchange packet must be encapsulated in a security packet to ensure that the appropriate security services are provided. IPsec and SUPERMAN both provide secure links between nodes, allowing for the secure exchange of data between end-points. Both, however, increase the size of the packets they secure, as they require headers and footers to provide authentication and integrity data, in addition to the confidentiality provided by encrypting the payload data.

Figure 7-11 shows the data costs associated with CBBA at layer 3 of the OS model (the network layer). CBBA task data (bundle exchange packets) is shown to allow for a comparative analysis of the security data costs of IPsec and SUPERMAN.

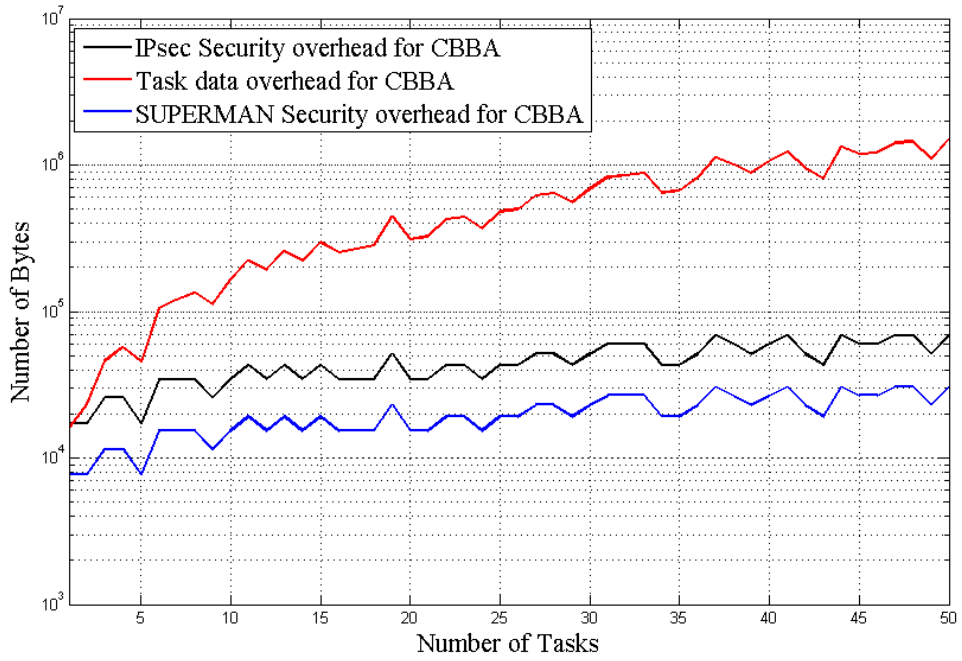

Figure 7-11 Graph showing the amount of data sent during CBBA DTA, including task data, SUPERMAN security data and IPsec security data requirement

The task data for CBBA in networks of 18 nodes ranges from 15,912 bytes (15.9 KB) for one task, to 1.5 MB for 50 tasks. This rapid rise in data requirements is demonstrated in the graph, with the relatively stable security costs of SUPERMAN and IPsec being dwarfed by the costs associated with sharing bundle data under CBBA.

IPsec is consistently more expensive than SUPERMAN in terms of security data cost. For one task, IPsec encapsulation actually generates more data than the bundles being sent, with 17.1 KB of data being required. For 50 tasks, only 68.5 KB of security data is required, 0.005% of the data cost of sharing bundles.

SUPERMAN requires 7.6 Kb of data to secure bundle sharing activities for 1 task. For 50 tasks, this increases to 30.6 KB. This is 0.002% of the load associated with bundle sharing for 50 task problems, and 44.6% of the overhead generated by IPsec. SUPERMAN required between 40 and 45% of the data needed by IPsec for all problem domain sizes simulated.

Figure 7-12 shows the overhead data requirements of CF-CBBA for bundle sharing and security using SUPERMAN and IPsec. Due to the more efficient communication of CF-CBBA, the total data requirement of bundle sharing and security is reduced.

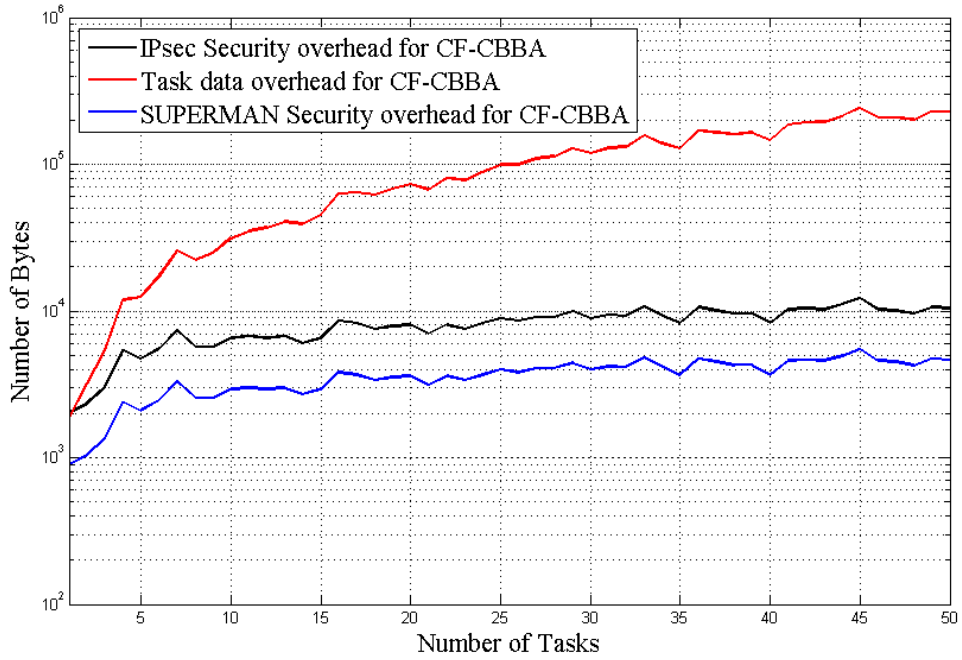

Figure 7-12 Graph showing the amount of data sent during CF-CBBA DTA, including task data, SUPERMAN security data and IPsec security data

CF-CBBA requires 1.8 KB of data to distribute one task. For problems involving 50 tasks, 228.4 KB is required. As with CBBA, CF-CBBA's task data requirements is consistently larger than the security overhead associated with either framework. This is expected, though for very small problem domains (1 or 2 tasks) IPsec requires more data than bundle sharing.

IPsec requires 2 KB of data to secure bundle exchange involving one task. More complex problems of 50 tasks require 10.4 KB of data. This is 4% of the data required for bundle exchange.

SUPERMAN, as previously shown, consistently requires less data to secure bundle exchange than IPsec, due to smaller packet size. 900 bytes are required to secure bundle exchanges involving 1 task, while larger problems of 50 tasks require an average of 4.6

KB. This is 45% of the data required by IPsec and 1.9% of the data sent to share bundles between nodes without security.

In each case, IPsec and SUPERMAN represent an additional cost on top of bundle sharing. SUPERMAN demonstrates that its smaller packet size, drawing on the network-based provision of security instead of IPsec's session oriented end-to-end provision of security, results in a lower security overhead. SUPERMAN provides authentication, confidentiality and integrity services over multi-hop routes, for a lower data cost than IPsec.

7.3.5 Section Summary

Modelling of SUPERMAN, IPsec and two secure routing protocols has been undertaken to provide data regarding the amount of data required to form a full network topology (routes between all nodes) under differing security frameworks and protocols. The use of reactive and proactive routing protocols allows for a comparative analysis of two common approaches to topology generation and maintenance in MANETs, and the effects of security on the data requirements of those processes.

Proactive protocols have been found to require less data due to the small packet sizes used in protocols like OLSR, and the simple algorithms used to generate a network topology under such protocols indicates that for a given instance of routing, such protocols are relatively lightweight and the addition of security, though incurring an additional cost in terms of packet size, retains this simplicity. As a result, proactive protocols, for individual instances of routing, were found to be faster than reactive protocols under SUPERMAN and IPsec.

The initialisation and security association services of SUPERMAN and the IKE and security association phase of IPsec have been quantified in terms of communication events and data requirement.

The additional throughput to be expected by using these required security control services has also been simulated. In both cases, SUPERMAN was found to require fewer communication events, and less data, than IPsec. The main advantage of SUPERMAN is

observed in its ability to allow any legitimate member node to perform the role of network administrator when authenticating new nodes or passing on credentials of nodes they have already authenticated. SUPERMAN provides a modular means of securing MANETs, targeted to reduce the cost of establishing security in such networks in the interests of being feasible to use in situations where the communication medium may be resource constrained or unreliable.

SUPERMAN has been profiled in terms of data requirements for the transmission of security credentials for initialisation and security association phases of the framework. This profile has been created by simulating SUPERMAN with three different tiers of security, forming low, medium and high security brackets. It has been shown that, as expected, increased key size will drive up the cost of communication in terms of the data communicated across the network to achieve a fully secured state.

These values allow the definition of critical framework attributes, forming a basis upon which further analysis can be performed and guidelines can be defined to determine the suitability of a given bracket of security for a target MANET based on resource constraints and security requirements.

The next section will analyse these results and provide qualitative observations of the results provided thus far, allowing for in-depth discussion of critical attributes.

7.4 Analysis of Results

7.4.1 Analysing the Provision of Security Services

The focus of modelling SUPERMAN and comparing it with IPsec, SAODV and SOLSR, was to provide quantitative data regarding the amount of data and number of communication events required to provide critical network services, including network and control services (routing and DTA). When considering the costs associated with security, however, the security services being offered must be taken into account.

Table 7-2 provides a comparison of the security services offered by the frameworks and protocols investigated in this work. The security dimensions outlined by the ITU-T X.805 are used as a basis for comparison.

SUPERMAN offers all services, with end-to-end and point-to-point provision of authenticated confidentiality and authenticated integrity respectively. By fully closing the network, SUPERMAN prevents trivial observation of packets and easy entry into the network.

Table 7-2 Security Feature Comparison

Dimensions	Security Protocol			
	SUPERMAN	SOLSR	SAODV	IPsec
Access Control	☑			☑
Authentication	☑			☑
Non-repudiation	☑			☑
Confidentiality	☑			☑
Communication Security	☑	☑	☑	☑
Data Integrity	☑	☑	☑	☑
Availability	☑	☑	☑	
Privacy	☑			☑
End-to-end	☑	☑	☑	☑
Point-to-point	☑	☑	☑	

SOLSR and SAODV provide end-to-end and point-to-point protection of routing packets, but do not offer privacy, confidentiality, non-repudiation, authentication or access control services. The focus of these protocols is to secure the routing process by preventing the manipulation or redirection of routing packets in a malicious manner.

IPsec provides all services but availability services and point-to-point. It only operates in an end-to-end manner, disregarding the route to allow private, secure communication between nodes irrespective of intermediaries. When considering the cost of security, the services obtained for that expense should be taken into account to reflect the benefits of paying the identified cost.

7.4.2 Analysis of Simulation Results

The following sub-sections analyse the results of simulation. The total cost of security is analysed to highlight the reduced cost of SUPERMAN security set-up for a whole MANET, when compared with IPsec. Secure routing is discussed to show the data cost of providing security services to routing protocols, specifically showing the costs incurred in addition to the standard data costs associated with the routing protocols in question. The cost of securing bundle exchange messages for consensus-based DTA is analysed to allow for discussion of the additional overheads required by IPsec and SUPERMAN to secure a vital control service needed to allow autonomous functionality in a MANET.

7.4.2.1 Analysing the Total Control Cost of Security

The total cost of security is the sum of initialisation and security association phases for SUPERMAN, and IKE and security association phases for IPsec. In this analysis, two key variables are analysed; communication events and data required to secure the network. Analysis is performed by observing the proportion of IPsec communication required by SUPERMAN to perform a similar role, and discussing in the context of the constraints likely to present themselves in an autonomous MANET with a variable number of nodes.

Figure 7-13 shows the percentage of IPsec communication events required by SUPERMAN to fully secure the network. Sub-section 7.3.1 clearly demonstrated the large difference between IPsec and SUPERMAN in terms of required communication events to provide a fully secured MANET, with the proportional difference shown below further highlighting this crucial differentiating factor.

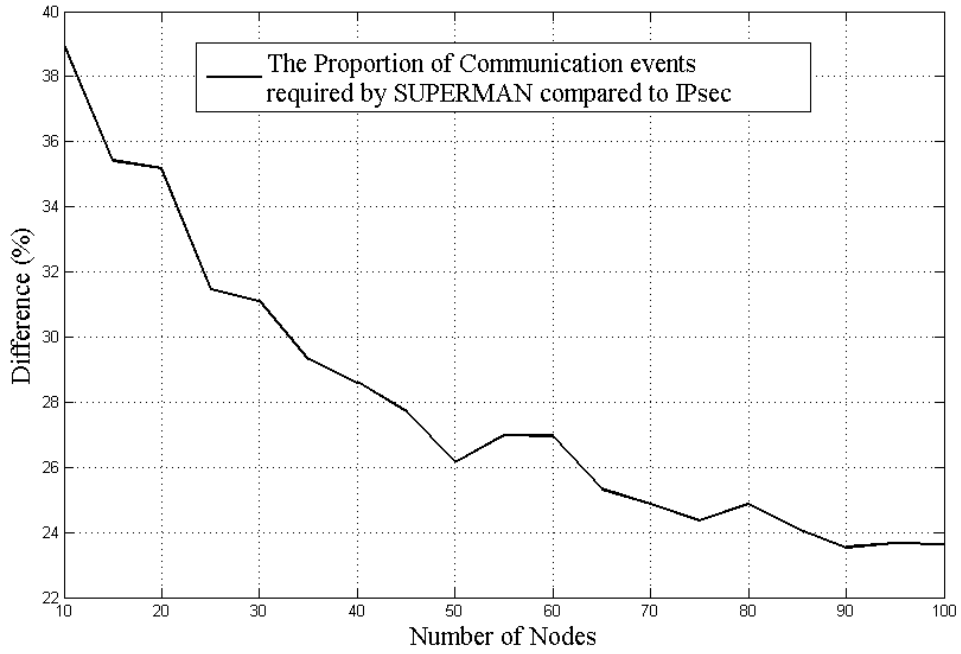

Figure 7-13 Graph showing the proportion of communication events when using SUPERMAN, compared to IPsec

For small networks, SUPERMAN requires an average of 38.8% of the communication events needed by IPsec. This is just over a third of the communication needed by IPsec to fully secure a network of 10 nodes. The gap between the two frameworks rapidly increases, with networks of 50 nodes requiring that SUPERMAN only send 26.2% of the messages needed by IPsec. For large MANETS of 100 nodes, SUPERMAN only requires 23.8% of the communication events required by IPsec.

SUPERMAN is designed with MANET constraints in mind. It is specifically calibrated to provide closed-network security (a VCN environment) at as low a cost as is possible. This includes one-time network access control and security association referral mechanisms. IPsec, though providing a similar level of security between end-points, has a much higher cost of set up as it must secure links between all nodes, instead of allowing the network to manage the secure association of nodes with the network and each other.

Figure 7-14 provides the proportional of data required by SUPERMAN to secure a MANET, when compared with IPsec. Exactly as was reported in Sub-section 7.3.4, two keys sizes have been analysed; 1024-bits and 2048-bits. These two key sizes represent

two levels of cryptographic complexity, with the higher value representing a larger, but potentially (mathematically) more secure key.

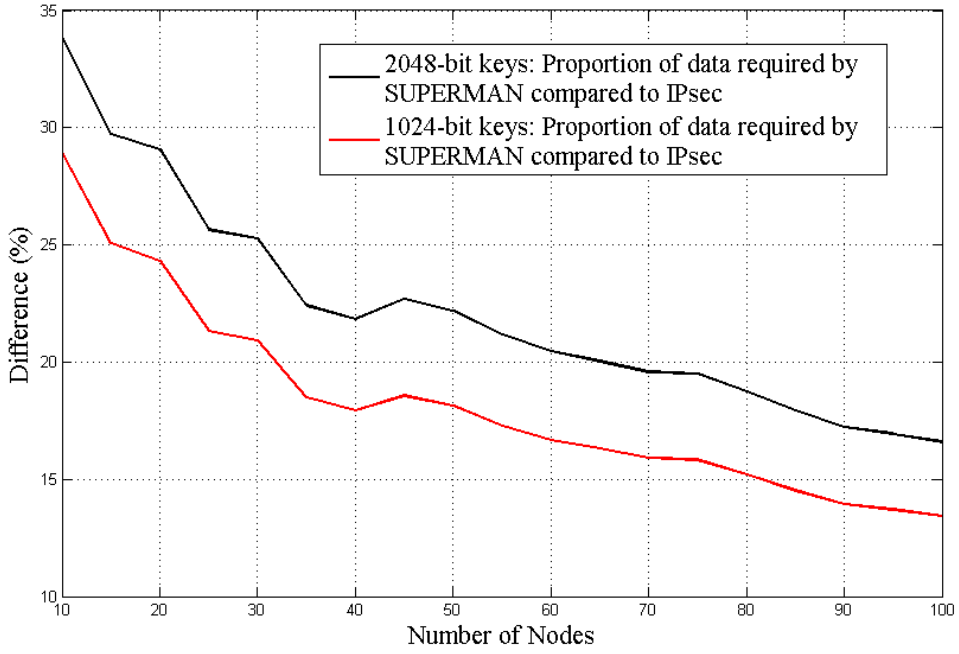

Figure 7-14 Graph showing the proportion of total data required IPsec, used by SUPERMAN

Bother sets of results demonstrate that SUPERMAN is consistently and significantly less data-intensive than IPsec. When using 1024-bit keys, SUPERMAN requires 28.4% of the data required by IPsec. For large MANETs of 100 nodes, SUPERMAN uses only 14% of the data needed by IPsec. In more cryptographically complex implementations using 2048-bit keys, SUPERMAN requires 34.6% of the data required by IPsec for 10 node networks. Large 100 node networks require SUPERMAN to send an average of 16.2% of the data required by IPsec.

The observations made regarding communication events hold for these figures. The MANET focus of SUPERMAN reduces the communication overhead of setting up a network-wide secure environment. One-time network access control and referred authentication and association of nodes greatly reduce the data costs associated with securing a MANET. As the size of the network increases, it becomes apparent that

SUPERMAN is more scalable than IPsec, due to the aforementioned features, as it requires less of the communication needed by IPsec as the network size increases.

7.4.2.2 Secure Routing

Sub-section 7.3.3 showed results of simulation for two MANET routing protocols, their secure implementations and a SUPERMAN secured version of each. This sub-section focuses on the security costs, specifically data requirements, incurred at the network layer by secure routing protocols and their SUPERMAN equivalents.

In each case, the cost of routing has been subtracted from the secure routing protocol and SUPERMAN results, to leave the cost of security. This allows for analysis of the relative security costs associated with each approach, showing the additional data requirements in addition to the total cost of forming routes between all nodes, as shown in Sub-section 7.3.3.

Figure 7-15 shows the data cost of securing AODV, using SAODV and SUPERAODV. As previously discussed, SAODV packets require multiple hashed fields, increasing the size of the packet considerably when compared to baseline AODV. These fields are required to provide integrity and authentication services, ensuring that the route formed only includes legitimate nodes. For the purpose of this analysis, it is assumed that no trust metrics are used, nodes are assumed to be trustworthy if they pass authentication, but must prove that they are legitimate members of the network. SUPERMAN adds a SUPERMAN header and a HMAC tag as a footer to each packet it encapsulates, but provides smaller packets than SAODV. Both approaches follow the same underlying protocol, with no behavioural differences.

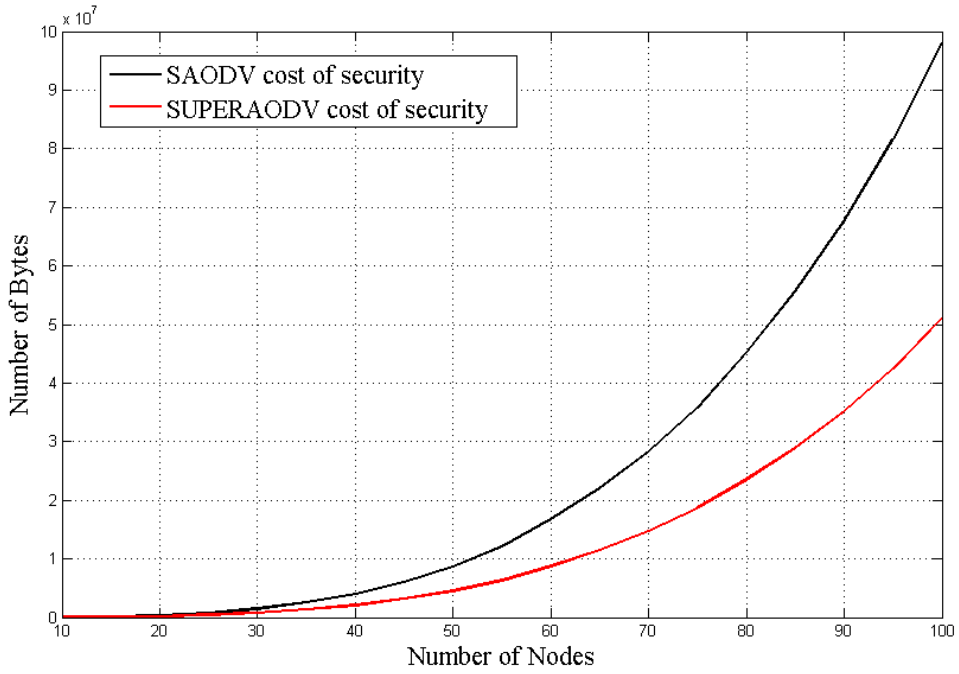

Figure 7-15 Graph showing the additional overhead cost of SOADV and SUPERAODV secure routing

SUPERAODV is shown to consistently require less data than SAODV to securely form routes between all nodes. This difference becomes significant in networks with more than 40 nodes, with SUPERAODV requiring only 62% of the data needed by SAODV for 50 nodes. For large networks of 100 nodes, SUPERAODV must send an average of 50.8 MB of additional security data, while SAODV must send 98.3 MB.

This difference, although large, is due to packet size only. There are no behavioural differences between the two protocols, with SUPERAODV following the same behaviour as SAODV, which in turn has the same behaviour as AODV at the network layer. As noted in Sub-section 7.3.3, secure routing only forms routes securely, it does not secure data travelling along that route. Therefore the additional security costs shown above are the security overhead of route formation, not the total security overhead of providing a secured route. The route may be trusted, but data must still be secured between end-points to prevent casual observation and potential modification by third parties.

Figure 7-16 shows the security overheads for SOLSR and SUPEROLSR. Two factors contribute to the difference between these two approaches; packet size and behavioural

differences. SOLSR requires replies from relay nodes to guarantee routing packet delivery and affirm the authenticity of nodes (and thus the routes formed over them). SUPEROLSR follows the behaviour of OLSR, a simple flood of routing packets designed to passively generate the network topology periodically.

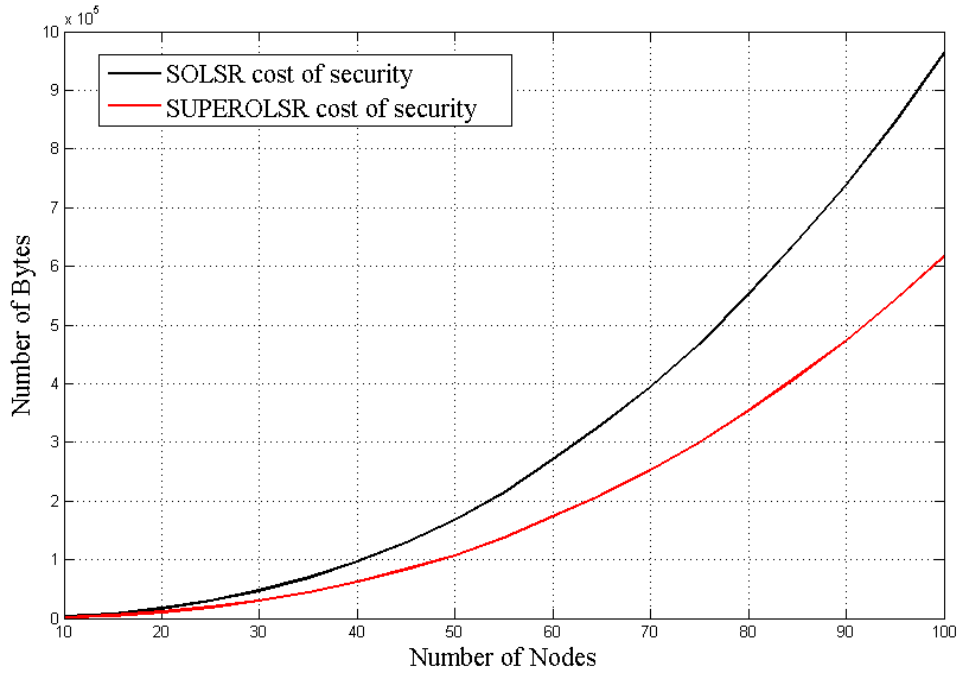

Figure 7-16 Graph showing the additional overhead cost of SOLSR and SUPEROLSR secure routing

SUPEROLSR, due to its simpler behaviour and smaller packet size, has consistently lower overhead than SOLSR. In networks of 30 or more nodes, this becomes significant, with 40 node networks showing SUPEROLSR overheads as 42% smaller than SOLSR overheads on average. For networks of 100 nodes, SUPEROLSR generates an additional security overhead of 609 KB, while SOLSR generates an overhead of 979.2 KB. In such networks, SUPEROLSR generates 37.8% less overhead than SOLSR.

Despite using the simpler, unsecured behaviour of OLSR (as it just encapsulates OLSR data, it does not modify routing behaviour) SUPEROLSR provides confidentiality, integrity and authentication services to all routing packets sent by members of the VCN. As a result, SUPEROLSR provides the security offered by SOLSR, for a lower cost at the point of forming routes. Sub-section 7.4.2.1 highlights the additional costs required

to set up a VCN under SUPERMAN however, and such costs are a pre-requisite for extending SUPERMAN security to any routing protocol.

7.4.2.3 Secure DTA

It has been established that although secure routing protects communication involving the formation of routes, it does not protect the data sent over those routes. As a result, the security overhead of control communication must be analysed. DTA (specifically consensus-based DTA in the context of this research) is required to grant autonomy to a MANET. Without task allocation algorithms, such networks are unable to form decisions and allocate nodes to specific duties. As a result, DTA is considered a vital control service that requires network-wide communication, thereby making it an ideal candidate for observing the additional security overheads associated with protecting such communication.

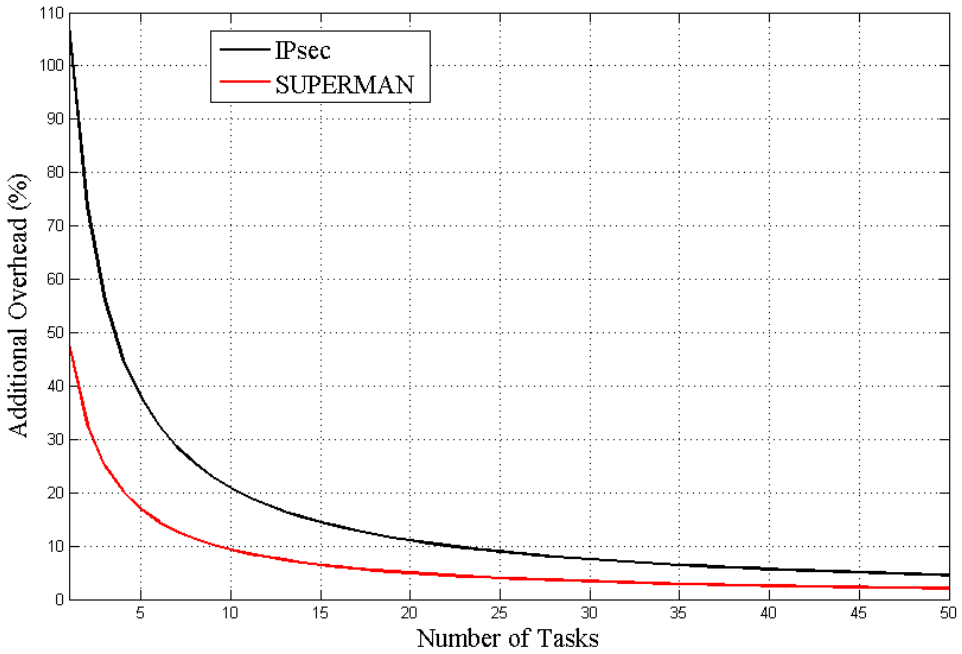

Figure 7-17 Graph showing the additional data cost of securing bundle exchange using IPsec and SUPERMAN

Figure 7-17 shows the security overheads for SUPERMAN and IPsec when securing DTA packets. As security overhead is tied directly to the number of communication events, acting as an additional number of bytes sent for every event, the additional overhead is identical for both CF-CBBA and CBBA.

The data shown is the percentage of the payload added in security bytes. This shows the proportion of additional security, relative to the payload it is protecting, to give an idea of the cost of security for a problem of a given size.

IPsec is shown to be excessively large for small problems, requiring 107% more data to secure bundle sharing communication involving 1 task. SUPERMAN is far less resource intensive, but still requires an additional 47% of the payload size in security data.

Both approaches rapidly improve as the problem domain increases in complexity. Problems involving 10 tasks involve IPsec appending an additional 20% of payload size in security data, while SUPERMAN appends an additional 9.5%. For 50 tasks, IPsec requires an additional 4.8% of payload size while SUPERMAN requires 1.6%.

Assuming that packets do not fragment (due to excessive payload size), this cost is constant. Fragmentation of a packet will double the security overhead. In this simulation, no packets fragmented, resulting in a stable decrease in security overhead as the problem domain complexity increased in size. SUPERMAN requires an average of 56.36% less data to secure bundle exchange than IPsec for problem domains of any size.

7.4.3 Section Summary

SUPERMAN has been investigated through comparative analysis, using IPsec as an equivalent framework for the purpose of comparison. The security cost of each framework has been modelled and simulated, focusing on the number of communication events required to achieve a fully secured network and the amount of data required to provide that service.

SUPERMAN's referral mechanism has been identified as the primary cost-reducing element of the framework. By allowing authenticated nodes to vouch for nodes that they

have had contact with previously, security association-related communication can be minimised, by reducing the amount of data that must be relayed, and the time taken to do so. By allowing all authenticated nodes to act as access controllers too, the process of joining the network may also avoid such repetition of communication, but allowing nodes to join the network over one hop of communication.

This also allows nodes not authenticated with the network to be closed out of routing operations, closing them against potential misbehaviour by nodes undergoing authentication. This shows that the proposed security framework provides a virtually closed network in all planes, securing network, control and client planes against any outside participation.

SUPERMAN has been shown to provide relatively lightweight data security, while requiring additional security control communication to establish a virtually closed network. Measures have been put in place to minimise this control traffic, the aim being to decrease the security overhead placed on limited-resource networks, specifically autonomous MANETs.

7.5 Chapter Summary

A test plan for modelling and simulating the SUPERMAN framework has been proposed. Comparison with IPsec has been undertaken to determine the relative characteristics of these two frameworks in terms of number of communication events and data required to provide a fully secured network. SUPERMAN has been shown to provide such an environment at a lower communication cost.

The cost of adding security to MANET routing protocols has been analysed. SAODV and SOLSR provide a mix of cryptographic and behavioural approaches to security, while SUPERMAN provides a purely cryptographic and service-based security solution. SUPERMAN has been shown to provide lower cost security than SAODV and SOLSR implementations of either.

The cost of adding security to network and control packets has been analysed, finding that SUPERMAN incurs less additional cost than IPsec in terms of additional bytes

required to secure DTA-related traffic. This was achieved by comparing its application to both CBBA and CF-CBBA with IPsec under the same conditions.

The next chapter will conclude the thesis, providing a summary of content provided throughout and identifying how original contributions have been met.

8 CONCLUSION

8.1 Chapter Introduction

This chapter concludes the thesis, documenting the original contributions that have been generated, and how the research gaps have been identified and addressed. Potential directions for future work are identified, to provide an outline of work that can follow on from that undertaken here and demonstrate potential future applications of the research undertaken to date.

8.1.1 Chapter Layout

- Section 8.2 documents the original contributions outlined in this thesis, outlining what those contributions are and providing a rationale for how they have been met.
- Section 8.3 reports on limitations in the research and critiques the research scope.
- Section 8.4 discusses future work, outlining recommended routes of further inquiry.
- Section 8.5 summarises the chapter and provides the final statement of the thesis.

8.2 Summary of Original Contributions

Section 1.5 identifies areas of original contribution, which have been addressed throughout this thesis. These original contributions are summarised in the following sub-sections. The original contribution will be identified, and the contributing proposals, tests and analyses that demonstrate the achievement of the contribution in question highlighted. This will provide evidence of novelty and the completion of research associated with the original contributions discussed in Chapter 1.

8.2.1 The proposal and analysis of Cluster Form CBBA, a method of clustering in CBBA to optimise communication

Chapter 4 documents a novel approach to clustering consensus-based task allocation algorithms, such as CBBA. The proposed approach is called Cluster Form CBBA (CF-CBBA), and it reduces the communication complexity of DTA using CBBA by sub-dividing the size of the network and problem domain to produce multiple simpler problems to solve. This is extended to include broadcast communication functionality in the BECF-CBBA proposal.

Chapter 5 tests CF-CBBA and BECF-CBBA, reporting on results and analysing the effectiveness of both protocols in reducing communication complexity. CF-CBBA was found to greatly reduce the complexity of CBBA-related communication, at the cost of having to plan clustering around the number of nodes and tasks present in a given problem. Failure to optimise assets to a given problem resulted in a loss of assignment optimality due to over-allocation of tasks to certain clusters. BECF-CBBA was shown to greatly reduce communication redundancy for both CBBA and CF-CBBA.

The proposal, testing and analysis of these DTA algorithms represents an original contribution achieved by this research.

8.2.2 The definition of Virtual Closed Networks, a means of providing VPN-like functionality to MANETs

Chapter 6 defines Virtual Closed Networks (VCN), networks with no physical boundaries to network access that collectively deny unknown parties access through a unifying access control and authentication policy. In traditional closed networks, hardware elements, such as firewalled routers, gate access to the network. MANETs cannot rely on topology control to force attackers to go through certain points of ingress, making traditional approaches ineffective.

By enacting a global access control policy across the MANET in question (such as that provided by SUPERMAN) it is possible to close a wireless MANET against intrusion. This harnesses the inherent duality of MANET nodes; they are both end-point and router, and so may enact router-focused security. In effect, each node is treated as a potential point of ingress and is required to uphold network security policies regarding networking authentication of nodes attempting to join their MANET.

The proposal of a VCN approach to MANET represents an original contribution resulting from the research that has been undertaken.

8.2.3 A vouching system (Security Using Pre-Existing Routing for Mobile Ad hoc Networks (SUPERMAN) referral mechanism) for key exchange to reduce the amount of communication required for multi-hop node authentication

The SUPERMAN referral mechanism, proposed in Chapter 6 and analysed in Chapter 7, represents a novel approach to mitigating the effects of route length on the number of transmissions required to facilitate node authentication in a MANET. Chapter 6 outlines the proposed mechanism, stating that by allowing members of a SUPERMAN network to vouch for each other, the length of routes may be reduced by having the first node with the credentials required by the source in its security table to respond on behalf of the destination node.

Chapter 7 provided results proving the efficacy of this approach, showing that the security association phase of SUPERMAN required significantly less information to be sent, when compared to IPsec. These findings, and the initial proposal, represent an original contribution.

8.2.4 Full-suite security for autonomous MANETs, in the form of the SUPERMAN framework

Chapter 6 proposed SUPERMAN, a security framework providing a full-suite of security features. All services outlined by the X.805 document are provided, as well as novel security protocols which augment the functionality of the framework in resource constrained MANETs.

Chapter 7 analysed the security services offered by SUPERMAN, IPsec, SAODV and SOLSR. It was found that SUPERMAN provided a full-suite of security feature, both end-to-end and point-to-point. SAODV and SOLSR only protect routing, with a limited set of security services. IPsec was found to provide a high level of security, but only end-to-end. As a result, SUPERMAN was found to provide a greater range of security services than comparable protocols and frameworks.

The SUPERMAN framework and associated findings represent a key original contribution of this research.

8.2.5 Performance analysis of secure routing, comparing SAODV, SOLSR and SUPERMAN

Chapter 7 documents the testing, report of results and subsequent analysis of SUPERMAN's security costs associated with secure routing. When compared to SAODV and SOLSR, SUPERMAN was found to provide lower cost security, by closing the network against the entry of potentially malicious nodes, instead of employing resource intensive behavioural methods such as those employed by SAODV and SOLSR due to an assumption of the open-medium problem remaining an issue during routing.

The reduction of security-cost for routing represents the completion of a key objective of the research, as routing is a continuous and potentially costly activity that must be performed throughout the lifetime of the network. The reduction of security costs associated with routing, by adopting a VCN approach to MANET security, represents an original contribution provided by this research.

8.2.6 Performance analysis of security-related control communication, comparing IPsec and SUPERMAN

Chapter 7 compares SUPERMAN and IPsec security costs associated with DTA. CBBA and CF-CBBA were used to profile the additional bytes required to secure DTA control data under both frameworks. SUPERMAN was found to outperform IPsec by a small margin, due to a smaller encapsulating packet size. This was found to be the result of SUPERMAN's focus on network-closure as a means of providing a persistently secure environment in which to communicate (via the VCN approach).

These findings represent an original contribution, as a product of the research documented in this thesis.

8.3 Limitations

Assumptions made in the formative stage of this research were discussed in Chapter 3. These assumptions present limitations in the research. These limitations represent the boundaries of the research, within which original contributions have been made. This section discusses those limitations, and potential studies and further work that may be undertaken, were the scope of the research extended.

8.3.1 Nodes are homogenous

All nodes are assumed to be identical, which is a reasonable assumption when considering simple surveying missions, but may not reflect the near-future deployment of drone swarms. Communication during task allocation could be affected by the introduction of nodes with different payloads or forms of mobility.

An example would be a network of UAVs and UGVs, as highlighted in Chapter 3, Sections 3.4 and 3.5. CF-CBBA, as proposed, would need to be extended with a fitting

function and capability-based clustering algorithm to account for networks with heterogeneous nodes.

8.3.2 Mobility is not modelled as a part of the simulations undertaken

Nodes are assumed to remain immobile during DTA, due to the calculation of distance between a node and a task being the basis for fitness to perform said task under CBBA. If nodes allocated tasks while moving significantly, there is a possibility that a solution may not be reached as the node considered most fit to perform a task may change every CBBA round.

When considering key exchange and authentication, it can be assumed that nodes will not move significantly during communication to require new routes to form. It is only when rerouting is required, that significant changes in communication cost would be observed, firstly due to lost packets when the route fails, then the additional traffic required to complete the operation once a new route has been generated.

The simulations used in this research could be extended with a mobility and topology-aware model as an item of future work, to allow for the appraisal of communication cost in networks of very fast moving nodes. These assumptions and supporting rationales are identified in Chapter 3, Section 3.4, Chapter 4, Section 4.4, and Chapter 6, Section 6.4.

8.3.3 Assumption of perfect channel performance

It is assumed, throughout this thesis, that the probability of delivery for a packet sent to a node in range is 1. This assumption has been made, due to the use of a variable loss rate having no effect on the cost of communication outside of accounting for retries. This would increase the costs associated with the baseline and novel algorithms being compared, but the relative difference between those algorithms would remain similar.

It would be beneficial, however, to use a realistic channel model for future work investigating the effects of packet loss on a full-system evaluation of the SUPERMAN

framework. Rerouting due to high loss rate on a route, and other scenarios would be worth consider when taking SUPERMAN to a prototype implementation stage.

8.3.4 Nodes are equipped with non-directional wireless transmitters

The simulations undertaken in this research assume a symmetrical, binary radio propagation model. Under such a model, radio propagation is perfectly spherical, and there is no degradation of transmission quality until the edge of transmission range is crossed. If a node is outside of the calculated range, it has a 100% loss rate. This model was chosen, because data-link and physical layers communication were not the focus of study in this research, and a simple channel model provided a reliable means of comparing DTA and security algorithm communication costs.

This could be improved in further work, by incorporating the algorithms used in this research, into a dedicated network simulator, such as OMNET++. This would allow for the study of elements not considered critical to the study reported in this thesis, such as the benefits of using certain transmitter hardware.

8.3.5 Constants, such as security credentials and task lists, are communicated prior to deployment

It is assumed that the Trusted Authority (TA) does not participate in the mission directly, and that all nodes have a shared origin (either the same TA, or a hierarchy including cooperating TAs). This has been found to be a reasonable assumption, based on previous literature and observations regarding the initialisation of nodes as a time of vulnerability, due to the sensitive data being shared with nodes by the TA at this time.

However, alternative approaches exist, such as distributed key negotiation and coalition-based assertion of authority in MANETs. It may be of value to consider such approaches as a potential replacement for the TA, as SUPERMAN only requires that an authority is able to provide nodes with certificates and identifying credentials at initialisation, not that

a single TA is required outside of the mission area. This would require further study of the vulnerabilities exposed by sharing fundamental data required to close the network in a potentially malicious environment.

8.4 Recommendations for Future Work

Throughout this thesis, the original contributions laid out in Chapter 1 and elaborated on in Chapters 2 and 3 have been addressed. However, the research undertaken has also highlighted areas of further potential contribution, which have fallen outside the scope of this research. The following areas of inquiry represent future work that could be undertaken, based on the proposals and findings presented in this document.

8.4.1 Context-aware Secure DTA Communication

Research into the requirement of task allocation functionality in autonomous MANETs has led to several observations about the relationship between the need to communicate information to involve all nodes in the allocation process, and the effective use of network resources for such communication. The addition of security increases the cost of any such communication by increasing packet size, by the length of required headers and HMAC tags in the case of SUPERMAN.

Future work to identify a means of applying contextual-awareness to security and task allocation functions would be useful in further increasing the efficiency with which an autonomous MANET communicates. By identifying which nodes needs to be involved in a given round of task allocation (possibly by identifying node capabilities and only involving qualifying nodes) the amount of communication between nodes on the network can be reduced.

Cluster size could also be considered in a context-aware system, accounting for size mismatch between clusters. Even if clusters are assumed to begin a mission with equal numbers of nodes, loss of nodes for various reasons may result in task allocations being

performed on mismatched clusters. Incorporating cluster size into a context-aware fitting function would allow cluster heads to bid on tasks appropriately, with respect to the capability of their cluster to service the identified tasks.

The level of security required to secure the communication may also be integrated into a context-aware system, allowing security levels to be assigned based on the current state of the mission area. In safe environments, footers containing security data may not be required, dramatically decreasing packet sizes.

The investigation and implementation of a context-aware autonomous MANET security algorithm, as an extension to SUPERMAN, is therefore identified as an item of future work.

8.4.2 Self-aware Distributed Task and Resource Management

The results of CF-CBBA simulation, when compared with CBBA, showed some interesting issues with assignment optimality. The use of arbitrary clusters was found to deliver sub-optimal assignments, networks divided into three clusters of six nodes would suffer form over allocation of tasks during the cluster-head allocation phase, leading to sub-optimal final assignments (as tasks cannot be outsourced once committed to a cluster level allocation). These results can be found in Chapter 5. Research into self-aware task allocation may lead to solutions to this optimality issue, allowing for expedient task allocation that retains all of the optimality of the original algorithm. It may also open up the path to adaptive task allocation, allowing networks to reassign tasks in response to local or global network change. This may extend to changes in the environment in which the network operates, and allow the implementation of missions with parameters that change based on events or the passage of time.

8.4.3 Investigating the effects of Topology on SUPERMAN Security

Chapter 7 notes the effects of SUPERMAN credential referral mechanisms of networks with many multi-hop routes. The simulations performed provided a range of 1 to 3 hops, but an average of only 1.23 hops in networks of 100 nodes due to the size of the simulation area. Due to the high proportion of neighbours to non-neighbouring nodes, the SUPERMAN referral mechanisms impact is not as high as it could hypothetically be.

Performing additional simulations, with a variety of use-cases, such as urban areas (obscured communication in small and large spaces forcing longer routes) and large-scale open-field simulations, would allow for the characteristics of SUPERMAN's referral mechanism to be analysed in greater depth. This would allow for further work to improve SUPERMAN or develop additional services to support and extend the efficient communication of security credentials while maintaining the reliability and integrity of the process.

8.4.4 Bridging SUPERMAN VCNs and other Networks

In a real world scenario, it is likely that a SUPERMAN VCN would not work in isolation. A particularly poignant scenario, is that of a disaster area in which search and rescue operations are making use of autonomous mobile assets to set up communication infrastructure and assist in the search operations.

In such a scenario, it is possible that the autonomous drones being used will serve a dual purpose. The first is to travel to targeted areas and provide information, regarding the state of the terrain or whether the area is populated. The second may be to facilitate communication with persons in that area, if telecommunications have been compromised by the disaster.

In such situations, the ability to form bridges between a SUPERMAN VCN and unsecured networks would be required. Means of providing such services could form the basis for substantial additional work. The use of IPsec to bridge over SUPERMAN VCNs is one possibility, another is providing facilities for SUPERMAN to bridge across

unsecured networks to other SUPERMAN VCNs, allowing for a highly distributed secure communication infrastructure to be deployed, which augments local unsecured telecommunications while preserving secure communications within the VCN themselves. As a result, this has been identified as an item of future work that is of particular interest to the researcher.

8.5 Chapter Summary

CF-CBBA and BECF-CBBA, two extensions to the CBBA DTA algorithm, have been shown to significantly reduce the communication costs associated with distributed task allocation. This has the effect of allowing larger problems to be solved using MANETs with limited network resources, potentially allowing small and inexpensive platforms to undertake mission independent of human control.

SUPERMAN, a novel security framework that focuses on the closure of a MANET against intrusion, has been proposed, tested and analysed. Low-cost, effective security has been a large part of the research objectives outlined in Chapter 1 and pursued throughout this thesis. As a part of this novel framework, the VCN approach to closing MANETs and other open-medium networks against intrusion has been proposed. SUPERMAN has been shown to outperform IPsec, SAODV and SOLSR in terms of additional security costs accrued by applying security to existing communication. SUPERMAN has also been found to significantly outperform IPsec when comparing the costs associated with setting up a secure network environment for communication, due to the proposal of a novel vouching system (the SUPERMAN referral mechanism).

This chapter has brought the thesis to a close, summarising the original contributions achieved by the research. Future work has been proposed, based on newly identified research gaps observed during the course of the research.

9 REFERENCES

Alani, M. M. (2010), 'Testing randomness in ciphertext of block-ciphers using diehard tests', *International Journal of Computer Science and Network Security* **10**(4), 53–57.

Ali, K. N., Basheeruddin, M., Moinuddin, S. K. & Lakkars, R. (2010), Manipsec-ipsec in mobile ad-hoc networks, *in* 'Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on', Vol. 1, IEEE, pp. 635–639.

Argyle, M., Casbeer, D. W. & Beard, R. (2011), A multi-team extension of the consensus-based bundle algorithm, *in* 'American Control Conference (ACC), 2011', IEEE, pp. 5376–5381.

Bade, S., Kumar, M. & Kamat, P. (2013), 'A reactive energy-alert algorithm for manet and its impact on node energy consumption', *International Journal of Computer Applications* **71**(18).

Bakshi, B., Krishna, P., Vaidya, N. & Pradhan, D. (1997), Improving performance of tcp over wireless networks, *in* 'Distributed Computing Systems, 1997., Proceedings of the 17th International Conference on', IEEE, pp. 365–373.

Baskett, F., Chandy, K. M., Muntz, R. R. & Palacios, F. G. (1975), 'Open, closed, and mixed networks of queues with different classes of customers', *Journal of the ACM (JACM)* **22**(2), 248–260.

Bellur, B., Lewis, M. & Templin, F. (2002), An ad-hoc network for teams of autonomous vehicles, *in* 'Proceedings of the First Annual Symposium on Autonomous Intelligence Networks and Systems', Citeseer.

Bethke, B., Valenti, M. & How, J. (2008), 'Uav task assignment', *Robotics & Automation Magazine, IEEE* **15**(1), 39–44.

Bhatia, J. & Shah, B. (2013), 'Review on various security threats & solutions and network coding based security approach for vanet', *International Journal of Advances in Engineering & Technology* **6**(1).

Botelho, S. & Alami, R. (1999), M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement, *in* 'Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on', Vol. 2, IEEE, pp. 1234–1239.

Brunet, L., Choi, H.-L. & How, J. P. (2008), Consensus-based auction approaches for decentralized task assignment, *in* 'AIAA Guidance, Navigation, and Control Conference, Honolulu, Hawaii'.

Bulygin, Y. (2007), Epidemics of mobile worms, *in* 'Performance, Computing, and Communications Conference, 2007. IPCCC 2007. IEEE Internationa', IEEE, pp. 475–478.

Burmester, M. & de Medeiros, B. (2009), 'On the security of route discovery in manets', *Mobile Computing, IEEE Transactions on* **8**(9), 1180–1188.

Chahidi, B. & Ezzati, A. (2012), 'Hybrid routing protocol for wireless sensor networks', *International Journal of Computer Science Issues (IJCSI)* **9**(2).

Chakeres, I. D. & Belding-Royer, E. M. (2004), Aodv routing protocol implementation design, *in* 'Distributed Computing Systems Workshops, 2004. Proceedings. 24th International Conference on', IEEE, pp. 698–703.

Chandra, A. (2005), 'Ontology for manet security threats', *PROC. NCON, Krishnankoil, Tamil Nadu* pp. 171–17.

Cheng, B.-N. & Moore, S. (2012), A comparison of manet routing protocols on airborne tactical networks, *in* 'Military Communications Conference, 2012-MILCOM 2012', IEEE, pp. 1–6.

Choi, H.-L., Brunet, L. & How, J. P. (2009), 'Consensus-based decentralized auctions for robust task allocation', *Robotics, IEEE Transactions on* **25**(4), 912–926.

Clausen, T. & Philippe, J. (2003), Optimized link state routing protocol (olsr), Technical report, No. RFC 3626.

Colistra, G., Pilloni, V. & Atzori, L. (2014), 'The problem of task allocation in the internet of things and the consensus-based approach', *Computer Networks* **73**, 98–111.

Comparetto, G., Schwartz, J., Schult, N. & Marshall, J. (2003), A communications analysis tool set that accounts for the attenuation due to foliage, buildings, and ground effects, *in* 'Military Communications Conference, 2003. MILCOM 2003. IEEE', Vol. 2, IEEE, pp. 1407–1411.

Dadhich, A., Gupta, A. & Yadav, S. (2014), Swarm intelligence based linear cryptanalysis of four-round data encryption standard algorithm, *in* 'Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on', IEEE, pp. 378–383.

Daranasi, S. K., Fatima, M. & Sharma, G. (2012), 'Performance analysis of manet routing protocols using three different mobility models', *Wireless Communication* **4**(14), 813–818.

Dasgupta, P. (2012), Multi-agent coordination techniques for multi-robot task allocation and multi-robot area coverage, *in* 'Collaboration Technologies and Systems (CTS), 2012 International Conference on', IEEE, pp. 75–75.

Deng, H., Li, W. & Agrawal, D. P. (2002), 'Routing security in wireless ad hoc networks', *Communications Magazine, IEEE* **40**(10), 70–75.

Dhanalakshmi, S. & Rajaram, M. (2008), 'A reliable and secure framework for detection and isolation of malicious nodes in manet', *IJCSNS* **8**(10), 184.

Doraswamy, N. & Harkins, D. (2003), *IPSec: the new security standard for the Internet, intranets, and virtual private networks*, Prentice Hall Professional.

Douceur, J. R. (2002), The sybil attack, *in* 'Peer-to-peer Systems', Springer, pp. 251–260.

Drucker, A., Kuhn, F. & Oshman, R. (2012), The communication complexity of distributed task allocation, *in* 'Proceedings of the 2012 ACM symposium on Principles of distributed computing', ACM, pp. 67–76.

Ducatelle, F., Förster, A., Di Caro, G. & Gambardella, L. (2009), 'Task allocation in robotic swarms: new methods and comparisons', *Dalle Molle Institute for Artificial Intelligence, Tech. Rep* .

Enneya, N., Oudidi, K. & Elkoutbi, M. (2009), 'Enhancing delay in manet using olsr protocol', *Int'l J. of Communications, Network and System Sciences* **2009**.

Garg, N. & Mahapatra, R. (2009), 'Manet security issues', *IJCSNS* **9**(8), 241.

Gerkey, B. & Mataric, M. (2003), Multi-robot task allocation: Analyzing the complexity and optimality of key architectures, *in* 'Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on', Vol. 3, IEEE, pp. 3862–3868.

Ghosh, A., Talpade, R., Elaoud, M. & Bereschinsky, M. (2005), Securing ad-hoc networks using ipsec, *in* 'Military Communications Conference, 2005. MILCOM 2005. IEEE', IEEE, pp. 2948–2953.

Glynos, D., Kotzanikolaou, P. & Douligeris, C. (2005), Preventing impersonation attacks in manet with multi-factor authentication, *in* 'Modelling and Optimization in Mobile, Ad Hoc, and Wireless Networks, 2005. WIOPT 2005. Third International Symposium on', IEEE, pp. 59–64.

Gu, D., Pei, G., Ly, H., Gerla, M., Zhang, B. & Hong, X. (2000), Uav aided intelligent routing for ad-hoc wireless network in single-area theater, *in* 'Wireless Communications and Networking Conference, 2000. WCNC. 2000 IEEE', Vol. 3, IEEE, pp. 1220–1225.

Gundry, S., Kusyk, J., Zou, J., Sahin, C. S. & Uyar, M. U. (2012), Performance evaluation of differential evolution based topology control method for autonomous manet nodes, *in* 'Computers and Communications (ISCC), 2012 IEEE Symposium on', IEEE, pp. 000228–000233.

Guo, Z., Sheikh, S., Al-Najjar, C., Kim, H. & Malakooti, B. (2010), 'Mobile ad hoc network proactive routing with delay prediction using neural network', *Wireless Networks* **16**(6), 1601–1620.

Haas, Z. J., Pearlman, M. R. & Samar, P. (2002), 'The zone routing protocol (zrp) for ad hoc networks', *draft-ietf-manet-zone-zrp-04.txt* .

Haddon, D. & Whittaker, C. (2003), 'Aircraft airworthiness certification standards for civil uavs', *Aeronautical Journal* **107**(1068 Spec), 79–86.

Hafslund, A., Tønnesen, A., Rotvik, R. B., Andersson, J. & Kure, Ø. (2004), Secure extension to the olsr protocol, *in* 'Proceedings of the OLSR Interop and Workshop, San Diego'.

Hinds, A., Ngulube, M., Zhu, S. & Al-Aqrabi, H. (2013), 'A review of routing protocols for mobile ad-hoc networks (manet)', *International Journal of Information and Education Technology* **3**(1).

Hunt, S., Meng, Q. & Hinde, C. (2012), An extension of the consensus-based bundle algorithm for multi-agent tasks with task based requirements, *in* 'Machine Learning and Applications (ICMLA), 2012 11th International Conference on', Vol. 2, IEEE, pp. 451–456.

Ivancic, W. D., Stewart, D. E., Sullivan, D. V. & Finch, P. E. (2012), *An evaluation of protocols for UAV science applications*, National Aeronautics and Space Administration, Glenn Research Center.

Jacquet, P., Muhlethaler, P., Clausen, T., Laouiti, A., Qayyum, A. & Viennot, L. (2001), Optimized link state routing protocol for ad hoc networks, *in* 'Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International', IEEE, pp. 62–68.

Jang, M. & Agha, G. (2006), 'Agent framework services to reduce agent communication overhead in large-scale agent-based simulations', *Simulation Modelling Practice and Theory* **14**(6), 679–694.

Jawandhiya, P. M., Ghonge, M. M., Ali, M. & Deshpande, J. (2010), 'A survey of mobile ad hoc network attacks', *International Journal of Engineering Science and Technology* **2**(9), 4063–4071.

Jayakumar, G. & Gopinath, G. (2007), 'Ad hoc mobile wireless networks routing protocols-a review', *Journal of Computer science* **3**(8), 574–582.

Jin, Y., Minai, A. A. & Polycarpou, M. M. (2003), Cooperative real-time search and task allocation in uav teams, *in* 'Decision and Control, 2003. Proceedings. 42nd IEEE Conference on', Vol. 1, IEEE, pp. 7–12.

Johnson, L. B., Choi, H.-L., Ponda, S. & How, J. P. (2012), Allowing nonsubmodular score functions in distributed task allocation, *in* 'IEEE Conference on Decision and Control (CDC), Dec', pp. 35–53.

Johnson, L. B., Ponda, S., Choi, H.-L. & How, J. P. (2010), Improving the efficiency of a decentralized tasking algorithm for uav teams with asynchronous communications, *in* 'AIAA Guidance, Navigation, and Control Conference (GNC)', Vol. 5, pp. 5406–5411.

Johnson, L. B., Ponda, S. S., Choi, H.-L. & How, J. P. (2011), Asynchronous decentralized task allocation for dynamic environments, *in* 'Proceedings of the AIAA Infotech Aerospace Conference', Vol. 2, pp. 2–14.

Kang, B.-H. & Balitanas, M. O. (2009), 'Vulnerabilities of vpn using ipsec and defensive measures', *International Journal of Advanced Science and Technology* **8**(9), 9–18.

Karim, S., Heinze, C. & Dunn, S. (2004), Agent-based mission management for a uav, *in* 'Intelligent Sensors, Sensor Networks and Information Processing Conference, 2004. Proceedings of the 2004', IEEE, pp. 481–486.

Kim, K.-S. & Choi, H.-L. (2014), A decentralized task allocation approach for cooperative transportation missions, *in* 'Control Automation Robotics & Vision (ICARCV), 2014 13th International Conference on', IEEE, pp. 1479–1483.

Kiran, P. S. (2009), 'Protocol architecture for mobile ad hoc networks', *2009 IEEE International Advance Computing Conference (IACC 2009)* .

Kumar, S., Pruthi, G., Yadav, A. & Singla, M. (2012), Security protocols in manets, *in* 'Advanced Computing & Communication Technologies (ACCT), 2012 Second International Conference on', IEEE, pp. 530–534.

Kuppusamy, P., Thirunavukkarasu, K. & Kalaavathi, B. (2011), A study and comparison of olsr, aodv and tora routing protocols in ad hoc networks, *in* 'Electronics Computer Technology (ICECT), 2011 3rd International Conference on', Vol. 5, IEEE, pp. 143–147.

Kusyk, J., Zou, J., Gundry, S., Sahin, C. S. & Uyar, M. U. (2013), 'Performance metrics for self-positioning autonomous manet nodes', *Journal of Cyber Security and Mobility* **2**(2), 151–173.

Lacey, T. H., Mills, R. F., Mullins, B. E., Raines, R. A., Oxley, M. E. & Rogers, S. K. (2012), 'Ripsec–using reputation-based multilayer security to protect manets', *computers & security* **31**(1), 122–136.

Lee, S.-J., Gerla, M. & Toh, C.-K. (1999), 'A simulation study of table-driven and on-demand routing protocols for mobile ad hoc networks', *Network, IEEE* **13**(4), 48–54.

Liu, Q., Zhang, D. & Zhao, Y. (2013), Study on framework of distributed key management for manets, *in* 'Information and Network Security (ICINS 2013), 2013 International Conference on', IET, pp. 1–6.

Lu, Q., Huang, W., Gong, X., Wang, X., Xiong, Y. & Miao, F. (2013), 'A secure distributed authentication scheme based on crt-vss and trusted computing in manet', *arXiv preprint arXiv:1307.2977* .

Maan, F. & Mazhar, N. (2011), Manet routing protocols vs mobility models: A performance evaluation, *in* 'Ubiquitous and Future Networks (ICUFN), 2011 Third International Conference on', IEEE, pp. 179–184.

Maity, S. & Ghosh, S. K. (2012), Enforcement of access control policy for mobile ad hoc networks, *in* 'Proceedings of the Fifth International Conference on Security of Information and Networks', ACM, pp. 47–52.

Matsumoto, M. & Nishimura, T. (1998), 'Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator', *ACM Transactions on Modeling and Computer Simulation (TOMACS)* **8**(1), 3–30.

McCune, R. R. & Madey, G. R. (2013), 'Swarm control of uavs for cooperative hunting with dddas', *Procedia Computer Science* **18**, 2537–2544.

McGee, A. R., Chandrashekhar, U. & Richman, S. H. (2004), Using itu-t x. 805 for comprehensive network security assessment and planning, *in* 'Telecommunications Network Strategy and Planning Symposium. Networks 2004, 11th International', IEEE, pp. 273–278.

Olfati-Saber, R. & Murray, R. M. (2004), 'Consensus problems in networks of agents with switching topology and time-delays', *Automatic Control, IEEE Transactions on* **49**(9), 1520–1533.

Ostergaard, E., Mataric, M. & Sukhatme, G. (2001), Distributed multi-robot task allocation for emergency handling, *in* 'Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on', Vol. 2, IEEE, pp. 821–826.

Papadimitratos, P. & Haas, Z. J. (2006), 'Secure data communication in mobile ad hoc networks', *Selected Areas in Communications, IEEE Journal on* **24**(2), 343–356.

Papadimitratos, P. P. & Haas, Z. J. (2003), 'Secure message transmission in mobile ad hoc networks', *Ad Hoc Networks Journal (Elsevier)* **1**(LCA-ARTICLE-2008-030), 193–209.

Patil, J. A. & Sidnal, N. (2013), 'Survey-secure routing protocols of manet', *International Journal of Applied Information Systems (IJAIS)* **5**(4).

Perkins, C. E. & Royer, E. M. (1999), Ad-hoc on-demand distance vector routing, *in* 'Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA'99. Second IEEE Workshop on', IEEE, pp. 90–100.

Phan, C. & Liu, H. H. (2008), A cooperative uav/ugv platform for wildfire detection and fighting, *in* 'System Simulation and Scientific Computing, 2008. ICSC 2008. Asia Simulation Conference-7th International Conference on', IEEE, pp. 494–498.

Pi, Y. (2011), 'A distributed consensus-based probabilistic auction algorithm for task assignment in multi-robot systems'.

Ponda, S. S., Johnson, L. B. & How, J. P. (2012), Distributed chance-constrained task allocation for autonomous multi-agent teams, *in* 'American Control Conference (ACC), 2012', IEEE, pp. 4528–4533.

Ponda, S. S., Johnson, L. B., Kopeikin, A. N., Choi, H.-L. & How, J. P. (2012), 'Distributed planning strategies to ensure network connectivity for dynamic heterogeneous teams', *Selected Areas in Communications, IEEE Journal on* **30**(5), 861–869.

Puttini, R., De Sousa, R. & Mé, L. (2004), Combining certification-based authentication and intrusion detection to secure manet routing protocols, *in* 'Proceedings of the 5th European Wireless Conference (Mobile and Wireless Systems beyond 3G)'.

Puttini, R. S., Me, L. & De Sousa, R. T. (2003), Certification and authentication services for securing manet routing protocols, *in* 'In Proceedings of the Fifth IFIP TC6 International Conference on Mobile and Wireless Communications Networks', Citeseer.

Quaritsch, M., Kruggl, K., Wischounig-Strucl, D., Bhattacharya, S., Shah, M. & Rinner, B. (2010), 'Networked uavs as aerial sensor network for disaster management applications', *e & i Elektrotechnik und Informationstechnik* **127**(3), 56–63.

Rachedi, A. & Benslimane, A. (2006), Trust and mobility-based clustering algorithm for secure mobile ad hoc networks, *in* 'Systems and Networks Communications, 2006. ICSNC'06. International Conference on', IEEE, pp. 72–72.

Reidt, S. & Wolthusen, S. (2008), Exploiting uavs capabilities in tactical manets, *in* 'Proceedings of the 2nd Annual Conference of ITA (AC-ITAâ€™08)', pp. 322–323.

Ren, W. & Beard, R. W. (2003), 'Dynamic consensus seeking in distributed multi-agent coordinated control'.

Ren, W. & Beard, R. W. (2005), 'Consensus seeking in multiagent systems under dynamically changing interaction topologies', *Automatic Control, IEEE Transactions on* **50**(5), 655–661.

Richard, A. O., Ahmad, A. & Kiseon, K. (2010), 'Security assessments of ieee 802.15. 4 standard based on x. 805 framework', *International Journal of Security and Networks* **5**(2), 188–197.

Royer, E. M. & Toh, C.-K. (1999), 'A review of current routing protocols for ad hoc mobile wireless networks', *Personal Communications, IEEE* **6**(2), 46–55.

Ryan, A., Zennaro, M., Howell, A., Sengupta, R. & Hedrick, J. (2004), An overview of emerging results in cooperative uav control, *in* 'Decision and Control, 2004. CDC. 43rd IEEE Conference on', Vol. 1, IEEE, pp. 602–607.

Saad, W., Han, Z., Basar, T., Debbah, M. & Hjorungnes, A. (2011), 'Hedonic coalition formation for distributed task allocation among wireless agents', *Mobile Computing, IEEE Transactions on* **10**(9), 1327–1344.

Saeed, N. H., Abbod, M. F. & Al-Raweshidy, H. S. (2012), Manet routing protocols taxonomy, *in* 'Future Communication Networks (ICFCN), 2012 International Conference on', IEEE, pp. 123–128.

Sen, J. (2010), A distributed trust and reputation framework for mobile ad hoc networks, *in* 'Recent Trends in Network Security and Applications', Springer, pp. 538–547.

Smith, D., Wetherall, J., Woodhead, S. & Adekunle, A. (2014), A cluster-based approach to consensus based distributed task allocation, *in* 'Parallel, Distributed and Network-Based Processing (PDP), 2014 22nd Euromicro International Conference on', IEEE, pp. 428–431.

Sommer, C. & Dressler, F. (2007), The dymo routing protocol in vanet scenarios, *in* 'Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th', IEEE, pp. 16–20.

Supriyanto, Hasbullah, I. H., Murugesan, R. K. & Ramadass, S. (2013), 'Survey of internet protocol version 6 link local communication security vulnerability and mitigation methods', *IETE Technical Review* **30**(1), 64–71.

Thanigaivel, G., Kumar, N. & Yogesh, P. (2012), Truncman: Trust based routing mechanism using non-cooperative movement in mobile ad-hoc network, *in* 'Digital Information and Communication Technology and it's Applications (DICTAP), 2012 Second International Conference on', IEEE, pp. 261–266.

Vincent, R., Berry, P., Agno, A., Ortiz, C. & Wilkins, D. (2003), Teambotica: a robotic framework for integrated teaming, tasking, networking, and control, *in* 'International Conference on Autonomous Agents: Proceedings of the second international joint conference on Autonomous agents and multi-agent systems', Vol. 14, pp. 1152–1153.

Von Mulert, J., Welch, I. & Seah, W. K. (2012), 'Security threats and solutions in manets: A case study using aodv and saodv', *Journal of network and computer applications* **35**(4), 1249–1259.

Wallgren, L., Raza, S. & Voigt, T. (2013), 'Routing attacks and countermeasures in the rpl-based internet of things', *International Journal of Distributed Sensor Networks* **2013**.

Wang, D. & Teng, J. (2013), 'Efficient and distributed authentication scheme for secure communication in manet', *Journal of Computational Information Systems* **9**(1), 57–64.

Yang, G., Guan, X. & Guo, Y. (2012), A location-aware parallel mac protocol for multihop wireless networks, *in* 'Information and Automation (ICIA), 2012 International Conference on', IEEE, pp. 636–639.

Yang, H., Luo, H., Ye, F., Lu, S. & Zhang, L. (2004), 'Security in mobile ad hoc networks: challenges and solutions', *Wireless Communications, IEEE* **11**(1), 38–47.

Zapata, M. G. (2002), 'Secure ad hoc on-demand distance vector routing', *ACM SIGMOBILE Mobile Computing and Communications Review* **6**(3), 106–107.

Zhao, Z., Hu, H., Ahn, G.-J. & Wu, R. (2012), 'Risk-aware mitigation for manet routing attacks', *Dependable and Secure Computing, IEEE Transactions on* **9**(2), 250–260.

Zhou, L. & Haas, Z. J. (1999), 'Securing ad hoc networks', *Network, IEEE* **13**(6), 24–30.

Zlot, R. & Stentz, A. (2006), 'Market-based multirobot coordination for complex tasks', *The International Journal of Robotics Research* **25**(1), 73–101.

Zou, C. C., Gong, W. & Towsley, D. (2003), Worm propagation modeling and analysis under dynamic quarantine defense, *in* 'Proceedings of the 2003 ACM workshop on Rapid malcode', ACM, pp. 51–60.

10 Appendices

10.1 Appendix A: DTA Simulation Code

The following code represents the files made or modified by the author of this thesis. Supporting code that has not been modified may be found at the following address: http://acl.mit.edu/projects/cbba.html . It is also available from the author of this document upon request.

A100runTestScript.m

```
% Test Script to control iteration count and output of data
% Authored by Darren Hurley-Smith, 2013. Last modified January 2015
% This test script runs for the defined number of iterations
% This test script requires the Parallel Processing Library for MATLAB

parpool('local',4);

rand_val = 11;
Agents = [];
Tasks = [];
AODV = [];
Data = [];
CBBA = [];
t_time=[];
c_time=[];
d_mess=[];
a_mess=[];
r_mess=[];
CBBA_count = 0;
Comms_count_h = [];
Comms_cluster = [];
cluster_time = [];
t_clust_score = [];

% Number of cluster heads
heads = 6;

% Number of cluster members
nodes = 3;

parfor x=1:100

    rand_val_t = rand_val*x;

    [Agent_Array, Task_Array, T_array, total_time, total_cluster_time,
comms_time, C_heads, C_cluster, total_heads_score, total_cluster_score,
```

```
Total_Tasks_Count, total_bytes] = costofcommunicationstestscript(rand_val_t,
heads, nodes);

    Agents(x,:)=Agent_Array;
    Tasks(x,:)=Task_Array;
    CBBA(x,:)=T_array;
    t_time(x,:)=total_time;
    c_time(x,:)=comms_time;
    t_bytes(x,:)=total_bytes;
    Comms_count_h(x,:)=C_heads+C_cluster;
    cluster_time(x,:)=total_cluster_time;

%     t_heads_score(x,:)=total_heads_score;
%     t_clust_score(x,:)=total_cluster_score;
end

avg_agents = mean(Agents(1:2,:),1);
avg_tasks = mean(Tasks(1:2,:),1);
avg_CBBA = mean(CBBA(1:2,:),1);

% total time including communications
avg_t_time=mean(t_time(1:2,:),1);

% total time for the average product of all clusters
avg_cluster_time=mean(cluster_time(1:2,:),1);

% communications time
avg_c_time=mean(c_time(1:2,:),1);

% cluster_heads comms
avg_Comms_count=mean(Comms_count_h(1:2,:),1);

avg_bytes=mean(t_bytes(1:2,:),1);

avg_Cluster_Comms=mean(Comms_cluster(1:2,:),1); %cluster comms
avg_t_heads_score=mean(t_heads_score(1:2,:),1);
avg_t_clust_score=mean(t_clust_score(1:2,:),1);

avg_CBBA=uint32(avg_CBBA);
avg_Comms_count=uint32(avg_Comms_count);
avg_Cluster_Comms=uint32(avg_Cluster_Comms);

delete(gcp);

% Uncomment below for automatic graph generation

% figure(1);

%plot tasks over total transmitted bits (plotting data and data+routing)
% plot(avg_tasks,avg_data,'-b');
% hold on;
% plot(avg_tasks,avg_AODV+avg_data,'-r');
% hold on;
```

```
% plot(avg_tasks,avg_AODV,'-k');

%Start plotting the total number of messages sent by type
% plot(avg_tasks,avg_d_mess,'-b');
% hold on;
% plot(avg_tasks,avg_a_mess,'-r');
% hold on;
% plot(avg_tasks,avg_r_mess,'-k');

%plot time taken to execute a given scenario
% plot(avg_tasks,avg_t_time,'-b');
% hold on;
% plot(avg_tasks,avg_t_time-avg_c_time,'-r');

% plot(avg_tasks,avg_t_time,'-b');
% hold on;
% plot(avg_tasks,avg_t_time+avg_c_time,'-r');
% hold on;
% plot(avg_tasks,avg_c_time,'-k');
% hold on;

% axis tight;
% xlabel ('Number of Tasks');
% xlabel ('# Tasks');
% ylabel ('Total Score');
% ylabel ('Time (seconds)');
% ylabel('End-to-End Communication Requests');
% ylabel ('Total Cost of Transmission (bits)');
% ylabel('Number of Messages Sent');
% ylabel ('CBBA Runs Count');
% grid on;
```

Costofcommunicationstestscript.m

```
% Copyright 2010
% Massachusetts Institute of Technology
% All rights reserved
% Developed by the Aerospace Controls Lab, MIT

% Extended by Darren Hurley-Smith, University of Greenwich, to include
multiple test graphing (May 2013), clustered DTA functionality (June 2013),
and support for CF-CBBA and BECF-CBBA simulation (July 2013)

%-------------------------------------------------------------------------%
% Consensus-Based Bundle Algorithm (CBBA)

% This software package implements the Consensus-Based Bundle Algorithm
% (CBBA), a decentralized market-based protocol that provides provably good
% approximate solutions for multi-agent multi-task allocation problems
% over networks of heterogeneous agents. The current version supports
```

```
% tasks with time windows of validity, heterogeneous agent-task
% compatibility requirements, and score functions that balance task
% reward and fuel costs.
%-------------------------------------------------------------------------%

%-------------------------------------------------------------------------%
% Main test file.  Initializes problem and calls CBBA.
%-------------------------------------------------------------------------%
function [Agent_Array, Task_Array, T_array,total_time, total_cluster_time,
comms_time,C_heads_array, C_cluster_array, total_heads_score,
total_cluster_score, Total_Tasks_Count, total_bytes] =
costofcommunicationstestscript(rand_val, H, N)
% profile on

SEED = rand_val;
rng(SEED,'twister');

% Clear environment
% close all; clear all;
addpath(genpath(cd));
Total_Tasks_Count=0;

% declare global variables and arrays
total_time=[];
comms_time=[];
Cluster_time_array=[];
total_cluster_time=[];
Task_Array = [];        % An array of tasks for graphing
CBBA_count_array = [];
d_count_array =[];
a_count_array=[];
r_count_array=[];
C_heads_array=[];
C_cluster_array=[];
C_cluster=[];
C_Total_Score=[];
Cluster_Total_Score=[];
total_heads_score=[];
total_cluster_score=[];
head_bytes=[];
cluster_bytes=[];
total_bytes=[];

totaltime = 0;
cluster_totaltime = [];

x = 1;                  % Initial number of tasks/agents (depending on set
variables)

%-------------------------------------------------------------------------%
% Initialize global variables
%-------------------------------------------------------------------------%
```

```
WORLD.CLR  = rand(100,3);

WORLD.XMIN = -2.0;
WORLD.XMAX =  2.5;
WORLD.YMIN = -1.5;
WORLD.YMAX =  5.5;
WORLD.ZMIN =  0.0;
WORLD.ZMAX =  2.0;
WORLD.MAX_DISTANCE = sqrt((WORLD.XMAX - WORLD.XMIN)^2 + ...
                          (WORLD.YMAX - WORLD.YMIN)^2 + ...
                          (WORLD.ZMAX - WORLD.ZMIN)^2);

%-------------------------------------------------------------------------%
% Define agents and tasks
%-------------------------------------------------------------------------%
% Grab agent and task types from CBBA Parameter definitions
CBBA_Params = CBBA_Init(0,0);

% Initialize possible agent fields
agent_default.id    = 0;            % agent id
agent_default.type  = 0;            % agent type
agent_default.avail = 0;            % agent availability (expected time in
sec)
agent_default.clr = [];             % for plotting

agent_default.clusterID = 0;      %identify cluster membership

agent_default.x       = 0;          % agent position (meters)
agent_default.y       = 0;          % agent position (meters)
agent_default.z       = 0;          % agent position (meters)
agent_default.nom_vel = 0;          % agent cruise velocity (m/s)
agent_default.fuel    = 0;          % agent fuel penalty (per meter)

% FOR USER TO DO:  Set agent fields for specialized agents, for example:
% agent_default.util = 0;

% Initialize possible task fields
task_default.id       = 0;          % task id
task_default.type     = 0;          % task type
task_default.value    = 0;          % task reward
task_default.start    = 0;          % task start time (sec)
task_default.end      = 0;          % task expiry time (sec)
task_default.duration = 0;          % task default duration (sec)
task_default.lambda   = 0.1;        % task exponential discount

task_default.x        = 0;          % task position (meters)
task_default.y        = 0;          % task position (meters)
task_default.z        = 0;          % task position (meters)

% FOR USER TO DO:  Set task fields for specialized tasks

%--------------------------%
```

```
% Create some default agents

% QUAD
agent_quad          = agent_default;
agent_quad.type     = CBBA_Params.AGENT_TYPES.QUAD; % agent type
agent_quad.nom_vel  = 2;         % agent cruise velocity (m/s)
agent_quad.fuel     = 1;         % agent fuel penalty (per meter)

% CAR
agent_car           = agent_default;
agent_car.type      = CBBA_Params.AGENT_TYPES.CAR;  % agent type
agent_car.nom_vel   = 2;         % agent cruise velocity (m/s)
agent_car.fuel      = 1;         % agent fuel penalty (per meter)

% Create some default tasks

% Track
task_track          = task_default;
task_track.type     = CBBA_Params.TASK_TYPES.TRACK;      % task type
task_track.value    = 100;    % task reward
task_track.start    = 0;      % task start time (sec)
task_track.end      = 100;    % task expiry time (sec) (set high for longer
missions)
task_track.duration = 5;      % task default duration (sec)

% Rescue
task_rescue          = task_default;
task_rescue.type     = CBBA_Params.TASK_TYPES.RESCUE;      % task type
task_rescue.value    = 100;    % task reward
task_rescue.start    = 0;      % task start time (sec)
task_rescue.end      = 100;    % task expiry time (sec)
task_rescue.duration = 15;     % task default duration (sec)


%-------------------------------------------------------------------------%
% Define sample scenario
%-------------------------------------------------------------------------%

% % while x<16, %uncomment to increment number of agents
% H = 5;      % # of cluster heads to be represented in total
% N = 6;      % # of agents in a cluster (including the cluster head)

% Create random agents
% create cluster heads in proportion to the number of clusters - this is a
% 'STATE' of an agent within a cluster, not a unique agent in itself
for h=1:H,
    X=[];
    Y=[];
    heads(h) = agent_quad;
    heads(h).id   = h;
    heads(h).clusterID = h;

    % declare cluster sizes and initialise agents for that cluster
```

```
    for n=1:N,
        agents(h,n) = agent_quad;
        agents(h,n).id    = rand(1);
        agents(h,n).x     = rand(1)*(WORLD.XMAX - WORLD.XMIN) + WORLD.XMIN;
        agents(h,n).y     = rand(1)*(WORLD.YMAX - WORLD.YMIN) + WORLD.YMIN;
        agents(h,n).clr   = WORLD.CLR(n,:);
        agents(h,n).clusterID = h;
    end

    %aggregate cluster position data for finding averages
    for n=1:N,
        X(n)=agents(h,n).x;
        Y(n)=agents(h,n).y;
    end

    %use cluster positions to determine 'virtual head' central position
    heads(h).x    = mean(X);
    heads(h).y    = mean(Y);
    heads(h).clr  = WORLD.CLR(h,:);
end

while x < 51, %uncomment for task increment - initialise to number of tasks
desired + 1
%     M = 25;     % # of tasks
    M=x;

    % Create random tasks
    for m=1:M,

        tasks(m) = task_track;

        tasks(m).id       = m;
        tasks(m).start    = rand(1)*100;
        tasks(m).end      = tasks(m).start + 1*tasks(m).duration;
        tasks(m).x        = rand(1)*(WORLD.XMAX - WORLD.XMIN) + WORLD.XMIN;
        tasks(m).y        = rand(1)*(WORLD.YMAX - WORLD.YMIN) + WORLD.YMIN;
        tasks(m).z        = rand(1)*(WORLD.ZMAX - WORLD.ZMIN) + WORLD.ZMIN;
    end

%----------------------------------------------------------------------%
% Initialize communication graph and diameter
%----------------------------------------------------------------------%

% Fully connected graph
    head_Graph = ~eye(H);   %comms graph for all cluster heads
    Graph = ~eye(N);        %comms graph for all other agents in a given
cluster

%----------------------------------------------------------------------%
% Run CBBA
%----------------------------------------------------------------------%
% do CBBA for cluster heads
```

```
    [CBBA_Assignments, Total_Score, T, heads_time, D, C_head_total,
[byte_count] = CBBA_Main(heads, tasks, head_Graph);
    C_heads_array(x) = C_head_total;
    total_heads_score(x) = Total_Score;
    total_time(x)= heads_time; %total cluster heads time to convergence
    head_bytes(x) = byte_count;

    airtime = 0.000416 + 3*(0.000032*numel(tasks));
    thistime = 0.000128 + airtime;
    totaltime = thistime+totaltime;
    comms_time(x)=totaltime;

% do CBBA for clusters
for h=1:H
    A = 0;
    for a=1:numel(CBBA_Assignments(h).bundle)
        for m=1:M
            if (tasks(m).id == CBBA_Assignments(h).bundle(a))
                % initialise cluster-tasks by allocating tasks from the
                % cluster-head bundle as a new task list
                cluster_tasks(a) = task_track;
                cluster_tasks(a).id       = tasks(m).id;
                cluster_tasks(a).start    = tasks(m).start;
                cluster_tasks(a).end      = tasks(m).end;
                cluster_tasks(a).x        = tasks(m).x;
                cluster_tasks(a).y        = tasks(m).y;
                cluster_tasks(a).z        = tasks(m).z;

                A=A+1; %counter of number of tasks in list
                Total_Tasks_Count=Total_Tasks_Count+1;
            end
        end
    end
    if (A>0)
        [CBBA_Cluster_Assignments, Total_Score, T, cluster_time, D, C_total,
byte_count] = CBBA_Main(agents(h,:), cluster_tasks, Graph);
        C_cluster(h)=C_total; %array of comm total arrays for cluster
        Cluster_time_array(h)=cluster_time;
        Cluster_Total_Score(h)=Total_Score;

        airtime = 0.000416 + 3*(0.000032*numel(cluster_tasks));
        thistime = 0.000128 + airtime;
        cluster_totaltime(x) = thistime;
        cluster_bytes(h) = byte_count;
        clear cluster_tasks;
    else
        C_cluster(h)=0; %array of comm total arrays for cluster heads
        Cluster_time_array(h)=0;
        Cluster_Total_Score(h)=0;
        cluster_bytes(h) = 0;
    end
end
```

```
    % comment out below to only include cluster-head comms
    comms_time(x)=comms_time(x)+(max(cluster_totaltime));

    %calculate total bytes sent
    total_bytes(x) = head_bytes(x) + (sum(cluster_bytes));

    C_cluster_array(x)=sum(C_cluster); %sum of all comm totals for cluster
heads

    total_cluster_time(x)=max(Cluster_time_array); %the slowest cluster time

%     total_heads_score(x)=C_Total_Score;
    total_cluster_score(x)=sum(Cluster_Total_Score); %sum of all cluster
scores represents total mission score
    %ignore heads score for the purposes of calculating the total mission
    %score


    Task_Array(x) = M;
    Agent_Array(x) = H; %placeholder for clusterhead count

    T_array(x) = T;
    x = x+1;

end
```

CBBACommunicate.m

```
% Copyright 2010
% Massachusetts Institute of Technology
% All rights reserved
% Developed by the Aerospace Controls Lab, MIT

% Modified by Darren Hurley-Smith (March 2013) to support Broadcast,
% Multicast and Unicast communication event tracking

%-----------------------------------------------------------------------%
% Runs consensus between neighbors
% Checks for conflicts and resolves among agents
%
% This is a message passing scheme described in Table 1 of:
% "Consensus-Based Decentralized Auctions for Robust Task Allocation",
% H.-L. Choi, L. Brunet, and J. P. How,
% IEEE Transactions on Robotics, Vol. 25, (4): 912 - 926, August 2009
%-----------------------------------------------------------------------%

function [CBBA_Data, t, D, C_count, byte_count] =
CBBA_Communicate(CBBA_Params, CBBA_Data, Graph, old_t, T, agents)

%set communications range
```

```
R = 10;
% k = 1; %this drone identity
airtime = 0;
thistime = 0;
totaltime = 0;
task_count = 0;
byte_count = 0;
% Copy data
for n = 1:CBBA_Params.N,
    old_z(n,:) = CBBA_Data(n).winners;
    old_y(n,:) = CBBA_Data(n).winnerBids;
end

z = old_z;
y = old_y;
t = old_t;

epsilon = 10e-6;


% Start communication between agents

% sender   = k REMOVED
% receiver = i NOW REPRESENTS NUMBER OF IDENTITIES INVOLVED
% task     = j still represents number of tasks

C_count=0;
task_count = numel(CBBA_Params.M);
C = cputime; %set timers for time analysis
D = 0;

for k=1:CBBA_Params.N

    C_count = C_count+1; %comment out if not in broadcast mode

%comment out below for broadcast comms (uncomment inner loop)
        byte_count = byte_count+(32+(numel(CBBA_Params.M)*16));

% comment out the abbove to release the need for identity communication -
% assume this has already been achieved for now
    for i=1:CBBA_Params.N
        %all communication happens here

        %comment out below for broadcast comms (uncomment above)
%         byte_count = byte_count+32+(task_count*16);

        %count the number of end to end communication events
        if i~=k %assume self-identity is equal to 1 at present
%             C_count = C_count+1;
        end

        if( Graph(k,i) == 1 ) %comment out to release requirement for
        %comms - graph isn't needed for self-contained allocation
```

```
            for j=1:CBBA_Params.M
                % Implement table for each task

                if( old_z(k,j) == k ) % Entries 1 to 4: Sender thinks he has
the task
                    task_count = task_count + 1;

                    % Entry 1: Update or Leave
                    if( z(i,j) == i )
                        if( old_y(k,j) - y(i,j) > epsilon )  % Update
                            z(i,j) = old_z(k,j);
                            y(i,j) = old_y(k,j);
                        elseif( abs(old_y(k,j) - y(i,j)) <= epsilon )  %
Equal scores
                            if( z(i,j) > old_z(k,j) )  % Tie-break based on
smaller index
                                z(i,j) = old_z(k,j);
                                y(i,j) = old_y(k,j);
                            end
                        end

                    % Entry 2: Update
                    elseif( z(i,j) == k )
                        z(i,j) = old_z(k,j);
                        y(i,j) = old_y(k,j);

                    % Entry 3: Update or Leave
                    elseif( z(i,j) > 0 )
                        if( old_t(k,z(i,j)) > t(i,z(i,j)) )  % Update
                            z(i,j) = old_z(k,j);
                            y(i,j) = old_y(k,j);
                        elseif( (old_y(k,j) - y(i,j)) > epsilon )  % Update
                            z(i,j) = old_z(k,j);
                            y(i,j) = old_y(k,j);
                        elseif( abs(old_y(k,j) - y(i,j)) <= epsilon )  %
Equal scores
                            if( z(i,j) > old_z(k,j) )  % Tie-break based on
smaller index
                                z(i,j) = old_z(k,j);
                                y(i,j) = old_y(k,j);
                            end
                        end

                    % Entry 4: Update
                    elseif( z(i,j) == 0 )
                        z(i,j) = old_z(k,j);
                        y(i,j) = old_y(k,j);

                    else
                        disp('Unknown winner value: Should not be here,
please revise')
                    end
```

```
                elseif( old_z(k,j) == i ) % Entries 5 to 8: Sender thinks
receiver has the task

                    % Entry 5: Leave
                    if( z(i,j) == i )
                        % Do nothing

                     % Entry 6: Reset
                    elseif( z(i,j) == k )
                        z(i,j) = 0;
                        y(i,j) = 0;

                     % Entry 7: Reset or Leave
                    elseif( z(i,j) > 0 )
                        if( old_t(k,z(i,j)) > t(i,z(i,j)) )  % Reset
                            z(i,j) = 0;
                            y(i,j) = 0;
                        end

                    % Entry 8: Leave
                    elseif( z(i,j) == 0 )
                        % Do nothing

                    else
                        disp('Unknown winner value: Should not be here,
please revise')
                    end

                elseif( old_z(k,j) > 0 ) % Entries 9 to 13: Sender thinks
someone else has the task

                    % Entry 9: Update or Leave
                    if( z(i,j) == i )
                        if( old_t(k,old_z(k,j)) > t(i,old_z(k,j)) )
                            if ( (old_y(k,j) - y(i,j)) > epsilon )
                                z(i,j) = old_z(k,j);  % Update
                                y(i,j) = old_y(k,j);
                            elseif( abs(old_y(k,j) - y(i,j)) <= epsilon )
                            % Equal scores

                                if( z(i,j) > old_z(k,j) )
                                % Tie-break based on smaller index

                                    z(i,j) = old_z(k,j);
                                    y(i,j) = old_y(k,j);
                                end
                            end
                        end

                     % Entry 10: Update or Reset
                    elseif( z(i,j) == k )
                        if( old_t(k,old_z(k,j)) > t(i,old_z(k,j)) )
```

```
                    % Update
                        z(i,j) = old_z(k,j);
                        y(i,j) = old_y(k,j);
                    else  % Reset
                        z(i,j) = 0;
                        y(i,j) = 0;
                    end

                % Entry 11: Update or Leave
                elseif( z(i,j) == old_z(k,j) )
                    if( old_t(k,old_z(k,j)) > t(i,old_z(k,j)) )
                    % Update

                        z(i,j) = old_z(k,j);
                        y(i,j) = old_y(k,j);
                    end

                % Entry 12: Update, Reset or Leave
                elseif( z(i,j) > 0 )
                    if( old_t(k,z(i,j)) > t(i,z(i,j)) )
                        if( old_t(k,old_z(k,j)) >= t(i,old_z(k,j)) )
                        % Update
                            z(i,j) = old_z(k,j);
                            y(i,j) = old_y(k,j);
                        elseif( old_t(k,old_z(k,j)) < t(i,old_z(k,j)) )
                        % Reset
                            z(i,j) = 0;
                            y(i,j) = 0;
                        else
                            disp('Should not be here, please revise')
                        end
                    else
                        if( old_t(k,old_z(k,j)) > t(i,old_z(k,j)) )
                            if( (old_y(k,j) - y(i,j)) > epsilon )
                            % Update

                                z(i,j) = old_z(k,j);
                                y(i,j) = old_y(k,j);
                           elseif( abs(old_y(k,j) - y(i,j)) <= epsilon )
                           % Equal scores

                                if( z(i,j) > old_z(k,j) )
                                % Tie-break based on smaller index

                                    z(i,j) = old_z(k,j);
                                    y(i,j) = old_y(k,j);
                                end
                            end
                        end
                    end

                % Entry 13: Update or Leave
                elseif( z(i,j) == 0 )
```

```
                        if( old_t(k,old_z(k,j)) > t(i,old_z(k,j)) )
                            % Update

                            z(i,j) = old_z(k,j);
                            y(i,j) = old_y(k,j);
                        end

                    else
                        disp('Unknown winner value: Should not be here,
please revise')
                    end

                elseif( old_z(k,j) == 0 )
                % Entries 14 to 17: Sender thinks no one has the task

                    % Entry 14: Leave
                    if( z(i,j) == i )
                        % Do nothing

                     % Entry 15: Update
                    elseif( z(i,j) == k )
                        z(i,j) = old_z(k,j);
                        y(i,j) = old_y(k,j);

                     % Entry 16: Update or Leave
                    elseif( z(i,j) > 0 )
                        if( old_t(k,z(i,j)) > t(i,z(i,j)) )  % Update
                            z(i,j) = old_z(k,j);
                            y(i,j) = old_y(k,j);
                        end

                    % Entry 17: Leave
                    elseif( z(i,j) == 0 )
                        % Do nothing

                    else
                        disp('Unknown winner value: Should not be here,
please revise')
                    end

                    % End of table

                else
                    disp('Unknown winner value: Should not be here, please
revise')
                end
            end

            % Update timestamps for all agents based on latest comm
            for n=1:CBBA_Params.N
                if( n ~= i && t(i,n) < old_t(k,n) )
                    t(i,n) = old_t(k,n);
                end
```

```
            end
            t(i,k) = T;
        end

    end
%     Uncomment to calculate approximate time taken to communicate based on
communication device specification
%     airtime = 0.000416 + 3*(0.000032*task_count);
%     thistime = 0.000128 + airtime;
%     totaltime = thistime+totaltime;
end
%
% D=(cputime-C)+totaltime+D;

% comment out top D for no comms, and bottom D for with commms
% D=(cputime-C)+totaltime;
% D=(cputime-C);
% Copy data
for n = 1:CBBA_Params.N,
    CBBA_Data(n).winners    = z(n,:);
    CBBA_Data(n).winnerBids = y(n,:);
end

% byte_count = sum(temp_byte);

end
```

10.2 Appendix B: Secure MANET Simulation Code

The following code allows the simulation of SUPERMAN, IPsec, SAODV, SOLSR, AODV and OLSR in a simple environment. It does not account for environmental affects, being a simple environment that tracks node placement and the routes generated between them.

Super_sim_setup.m

```
% Darren Hurley-Smith, University of Greenwich 2015
% Simulation set up script for SUPERMAN security cost evaluation
% Runs for a user defined number of iterations, over a user defined network
% size

function [SUPERMAN_comm_array, SUPERMAN_init_array, IP_tunnel_cost,
IP_init_cost, IParr_bytes, SUPERarr_bytes, IParr_init_bytes,
SUPERarr_init_bytes, SAODVresbytes, SOLSRresbytes, SUPERAODVresbytes,
SUPEROLSRresbytes, AODV_resbytes, OLSR_resbytes, n_count_array, t] =
SUPER_Sim_Setup(N, X, Y, Iterations, SEED)
```

```
% Start Tictoc. Use only to profile sim time, do not use as a result
tic;

parpool('local',4);

% Initialise rng with seed and set to chosen RNG type
rng(SEED,'twister');

% Set up required variables here
SUPERMAN_comm_array = [];
n_count_array = [];
SUPERMAN_init_array = [];
IParr_bytes = [];
SUPERarr_bytes = [];
IParr_init_bytes = [];
SUPERarr_init_bytes = [];
IP_tunnel_cost = [];
IP_init_cost = [];
comm_count = 0;
IPauth = 0;
IPinit = 0;
SUPER_bytes = 0;
IP_bytes = 0;
SUPER_init_bytes = 0;
IP_init_bytes = 0;
init_val = 0;
SAODV_prebytes = 0;
SOLSR_prebytes = 0;
SUPERAODV_prebytes = 0;
SUPEROLSR_prebytes = 0;
AODV_prebytes = 0;
OLSR_prebytes = 0;
SAODVresbytes = [];
SOLSRresbytes = [];
SUPERAODVresbytes = [];
SUPEROLSRresbytes = [];
AODV_resbytes = [];
OLSR_resbytes = [];
Dcount=0;
Ccount=0;
CEcount=0;
SKcount=0;
D_array=zeros(1,N-9);
CR_array=zeros(1,N-9);
CE_array=zeros(1,N-9);
SK_array=zeros(1,N-9);
nodes = [];
count = 1;

for n=10:5:N
    parfor i=1:Iterations
        [SUPERMAN_comm_count,
SUPERMAN_comm_count_init,IPtotal_comms,IPinit_comms, IPSEC_bytes,
```

```
tunnel_bytes, total_init_bytes, IPSEC_init_bytes, SAODV_bytes, SOLSR_bytes,
SUPERAODV_bytes, SUPEROLSR_bytes, AODV_bytes, OLSR_bytes] =
AuthSim_main(n,X,Y);
        comm_count = comm_count+SUPERMAN_comm_count;
        init_val = init_val+SUPERMAN_comm_count_init;
        IPauth = IPauth+IPtotal_comms;
        IPinit = IPinit+IPinit_comms;
        SUPER_bytes = SUPER_bytes+tunnel_bytes;
        IP_bytes = IP_bytes+IPSEC_bytes;
        SUPER_init_bytes = SUPER_init_bytes + total_init_bytes;
        IP_init_bytes = IP_init_bytes + IPSEC_init_bytes;
        SAODV_prebytes = SAODV_prebytes+SAODV_bytes;
        SOLSR_prebytes = SOLSR_prebytes+SOLSR_bytes;
        SUPERAODV_prebytes = SUPERAODV_prebytes+SUPERAODV_bytes;
        SUPEROLSR_prebytes = SUPEROLSR_prebytes+SUPEROLSR_bytes;
        AODV_prebytes = AODV_prebytes + AODV_bytes;
        OLSR_prebytes = OLSR_prebytes + OLSR_bytes;
        disp(sprintf('Iteration %d of %d completed',i,Iterations))
    end

    SUPERMAN_comm_array(count) = comm_count/Iterations;
    SUPERMAN_init_array(count) = init_val/Iterations;
    IP_tunnel_cost(count) = IPauth/Iterations;
    IP_init_cost(count) = IPinit/Iterations;
    SUPERarr_bytes(count) = SUPER_bytes/Iterations;
    IParr_bytes(count) = IP_bytes/Iterations;
    SUPERarr_init_bytes(count) = SUPER_init_bytes/Iterations;
    IParr_init_bytes(count) = IP_init_bytes/Iterations;
    SAODVresbytes(count) = SAODV_prebytes/Iterations;
    SOLSRresbytes(count) = SOLSR_prebytes/Iterations;
    SUPERAODVresbytes(count) = SUPERAODV_prebytes/Iterations;
    SUPEROLSRresbytes(count) = SUPEROLSR_prebytes/Iterations;
    AODV_resbytes(count) = AODV_prebytes/Iterations;
    OLSR_resbytes(count) = OLSR_prebytes/Iterations;
    n_count_array(count) = n;
    comm_count = 0;
    init_val = 0;
    IPauth = 0;
    IPinit = 0;
    IPauth_bytes = 0;
    IPinit_bytes = 0;
    disp(sprintf('Cluster %d of %d completed',n,N))

    count = count +1;
end

delete(gcp);

% Tictoc timing for simulation profiling only. DO NOT use for simulation
% data, not reflective of actual time to process/communicate
```

AuthSim_Main.m

```
%Darren Hurley-Smith, University of Greenwich 2015

%Main script for Network generation.

%Creates the world_space, node and network objects for further simulation.
%Nodes are randomly spaced in the world, within range of at least one other
%node.  All nodes are static during the authentication simulation process.

function [SUPERMAN_comm_count, SUPERMAN_comm_count_init, IPtotal_comms,
IPinit_comms, IPbytes, tunnel_bytes, total_init_bytes, IPSEC_init_bytes,
SAODV_bytes, SOLSR_bytes, SUPERAODV_bytes, SUPEROLSR_bytes, AODV_bytes,
OLSR_bytes, node] = AuthSim_main( N, X, Y)

% rand('seed', SEED);
% rng(SEED,'twister');

R = 100;
WORLD.XMIN = 0;
WORLD.YMIN = 0;
WORLD.XMAX = X;
WORLD.YMAX = Y;

farthestPreviousHop = [];
farthestNextHop = [];

routingTable = createArrays(N, [1 N]);
authTable = zeros(1,N);

fail_check = 1;

CostMatrix = [];

xy = [N 2];

    for n=1:N,
        node(n).Index = n;
        node(n).rTable = routingTable;
        node(n).aTable = authTable;
        node(n).IPaTable = authTable;
        node(n).x    = rand(1)*WORLD.XMAX;
        node(n).y    = rand(1)*WORLD.YMAX;
        node(n).z    = 1;
        node(n).netauth = 0; %set initial network authentication level to 0
until joined
        node(n).IPnetauth = 0;
        node(n).adjacent_nodes = 0;
    end

    node(1).netauth = 1; %initialise node 1 to allow propagation of network
details
```

```
    node(1).IPnetauth = 1; %same for IPSEC network auth

    %generate connection matrix for all nodes
    for n=1:N,
        for j = 1:N
            distance = sqrt((node(n).x - node(j).x)^2 + (node(n).y -
node(j).y)^2);
            if distance <= R
                matrix(n,j)=1;
                netCostMatrix(n,j)=(node(n).z + node(j).z)/2;
                matriz(n,j)=distance;
            else
                matrix(n,j) = inf;
                netCostMatrix(n,j)= inf;
                matriz(n,j)=inf;
                CostMatrix = netCostMatrix;
            end
            xy(n,:)=node(n).x;
            xy(:,n)=node(n).y;
        end
    end

    for i = 1:N
        % initialize the farthest node to be itself;
        farthestPreviousHop(i) = i;     % used to compute the RTS/CTS range;
        farthestNextHop(i) = i;
    end;
    %Initialise network
    for i=1:N
        for j=1:N
            if j ~= i
%             node(i).rTable{j}=AODV_routing(N, matrix, node(i).Index,
node(j).Index, farthestPreviousHop, farthestNextHop);
%             node(i).rTable{j} =
dijkstra(matrix,node(i).Index,node(j).Index);
            node(i).rTable{j} =
dijkstra_improved(matrix,xy,node(i).Index,node(j).Index);
            end
        end
    end

% Calculate adjacency for the purpose of calculating broadcasts

    for n=1:N
        for j=1:N
            if n ~= j
                if length(node(n).rTable{j})-1 == 1
                    node(n).adjacent_nodes = node(n).adjacent_nodes + 1;
                end
            end
        end
    end
```

```
% Calculate the cost of routing (bytes) for SAODV, SOLSR and SUPERMAN
% protected AODV and OLSR

[SAODV_bytes, node] = SAODV(node, N);
[SOLSR_bytes, node] = SOLSR(node, N);
[SUPERAODV_bytes, node] = SUPERAODV(node, N);
[SUPEROLSR_bytes, node] = SUPEROLSR(node, N);
[AODV_bytes, node] = AODV(node, N);
[OLSR_bytes, node] = OLSR(node, N);

%Perform initial authentication.  At this stage routes cannot be used.
%Source will only receive from or send to nodes one hop from itself to
%represent this

%SUPERMAN_comm_count_init = 0;

[SUPERMAN_comm_count_init, node, total_init_bytes] =
SUPERMAN_Init_Auth(node,N);

%Perform SUPERMAN authentication (assume that all nodes have authenticated
%with the network already and have a valid NSb (and resulting SKbe and SKbp
%keys)

[SUPERMAN_comm_count, node, tunnel_bytes] = SUPERMAN_Link_Auth(node, N);

%Now test IPSEC characteristics
[IPtotal_comms, node, IPbytes, IPinit_comms, IPSEC_init_bytes] =
IPSEC_Link_Auth(node, N);

% [IPinit_comms, node, IPSEC_init_bytes] = IP_Init_Auth(node, N);


end
```

Dijkstra_improved.m

(modified code from an external source, supplementary files available on request)

```
function [paths] = dijkstra_improved(AorV,xyCorE,SID,FID,iswaitbar)

%DIJKSTRA Calculate Minimum Costs and Paths using Dijkstra's Algorithm
%
%   Inputs:
%     [AorV] Either A or V where
%         A   is a NxN adjacency matrix, where A(I,J) is nonzero (=1)
%               if and only if an edge connects point I to point J
%               NOTE: Works for both symmetric and asymmetric A
%         V   is a Nx2 (or Nx3) matrix of x,y,(z) coordinates
%     [xyCorE] Either xy or C or E (or E3) where
%         xy  is a Nx2 (or Nx3) matrix of x,y,(z) coordinates (equivalent to
V)
```

```
%               NOTE: only valid with A as the first input
%         C   is a NxN cost (perhaps distance) matrix, where C(I,J) contains
%               the value of the cost to move from point I to point J
%               NOTE: only valid with A as the first input
%         E   is a Px2 matrix containing a list of edge connections
%               NOTE: only valid with V as the first input
%         E3  is a Px3 matrix containing a list of edge connections in the
%               first two columns and edge weights in the third column
%               NOTE: only valid with V as the first input
%     [SID] (optional) 1xL vector of starting points
%         if unspecified, the algorithm will calculate the minimal path from
%         all N points to the finish point(s) (automatically sets SID = 1:N)
%     [FID] (optional) 1xM vector of finish points
%         if unspecified, the algorithm will calculate the minimal path from
%         the starting point(s) to all N points (automatically sets FID =
1:N)
%     [iswaitbar] (optional) a scalar logical that initializes a waitbar if
nonzero
%
%   Outputs:
%     [costs] is an LxM matrix of minimum cost values for the minimal paths
%     [paths] is an LxM cell array containing the shortest path arrays
%
%   Revision Notes:
%     (4/29/09) Previously, this code ignored edges that have a cost of
zero,
%     potentially producing an incorrect result when such a condition
exists.
%     I have solved this issue by using NaNs in the table rather than a
%     sparse matrix of zeros. However, storing all of the NaNs requires more
%     memory than a sparse matrix. This may be an issue for massive data
%     sets, but only if there are one or more 0-cost edges, because a sparse
%     matrix is still used if all of the costs are positive.
%
%   Note:
%     If the inputs are [A,xy] or [V,E], the cost is assumed to be (and is
%       calculated as) the point-to-point Euclidean distance
%     If the inputs are [A,C] or [V,E3], the cost is obtained from either
%       the C matrix or from the edge weights in the 3rd column of E3
%
%   Example:
%       % Calculate the (all pairs) shortest distances and paths using
[A,xy] inputs
%       n = 7; A = zeros(n); xy = 10*rand(n,2)
%       tri = delaunay(xy(:,1),xy(:,2));
%       I = tri(:); J = tri(:,[2 3 1]); J = J(:);
%       IJ = I + n*(J-1); A(IJ) = 1
%       [costs,paths] = dijkstra(A,xy)
%
%   Example:
%       % Calculate the (all pairs) shortest distances and paths using [A,C]
inputs
%       n = 7; A = zeros(n); xy = 10*rand(n,2)
```

```
%       tri = delaunay(xy(:,1),xy(:,2));
%       I = tri(:); J = tri(:,[2 3 1]); J = J(:);
%       IJ = I + n*(J-1); A(IJ) = 1
%       a = (1:n); b = a(ones(n,1),:);
%       C = round(reshape(sqrt(sum((xy(b,:) - xy(b',:)).^2,2)),n,n))
%       [costs,paths] = dijkstra(A,C)
%
%   Example:
%       % Calculate the (all pairs) shortest distances and paths using [V,E]
inputs
%       n = 7; V = 10*rand(n,2)
%       I = delaunay(V(:,1),V(:,2));
%       J = I(:,[2 3 1]); E = [I(:) J(:)]
%       [costs,paths] = dijkstra(V,E)
%
%   Example:
%       % Calculate the (all pairs) shortest distances and paths using
[V,E3] inputs
%       n = 7; V = 10*rand(n,2)
%       I = delaunay(V(:,1),V(:,2));
%       J = I(:,[2 3 1]);
%       D = sqrt(sum((V(I(:),:) - V(J(:),:)).^2,2));
%       E3 = [I(:) J(:) D]
%       [costs,paths] = dijkstra(V,E3)
%
%   Example:
%       % Calculate the shortest distances and paths from the 3rd point to
all the rest
%       n = 7; V = 10*rand(n,2)
%       I = delaunay(V(:,1),V(:,2));
%       J = I(:,[2 3 1]); E = [I(:) J(:)]
%       [costs,paths] = dijkstra(V,E,3)
%
%   Example:
%       % Calculate the shortest distances and paths from all points to the
2nd
%       n = 7; A = zeros(n); xy = 10*rand(n,2)
%       tri = delaunay(xy(:,1),xy(:,2));
%       I = tri(:); J = tri(:,[2 3 1]); J = J(:);
%       IJ = I + n*(J-1); A(IJ) = 1
%       [costs,paths] = dijkstra(A,xy,1:n,2)
%
%   Example:
%       % Calculate the shortest distance and path from points [1 3 4] to [2
3 5 7]
%       n = 7; V = 10*rand(n,2)
%       I = delaunay(V(:,1),V(:,2));
%       J = I(:,[2 3 1]); E = [I(:) J(:)]
%       [costs,paths] = dijkstra(V,E,[1 3 4],[2 3 5 7])
%
%   Example:
%       % Calculate the shortest distance and path between two points
%       n = 1000; A = zeros(n); xy = 10*rand(n,2);
```

```
%       tri = delaunay(xy(:,1),xy(:,2));
%       I = tri(:); J = tri(:,[2 3 1]); J = J(:);
%       D = sqrt(sum((xy(I,:)-xy(J,:)).^2,2));
%       I(D > 0.75,:) = []; J(D > 0.75,:) = [];
%       IJ = I + n*(J-1); A(IJ) = 1;
%       [cost,path] = dijkstra(A,xy,1,n)
%       gplot(A,xy,'k.:'); hold on;
%       plot(xy(path,1),xy(path,2),'ro-','LineWidth',2); hold off
%       title(sprintf('Distance from 1 to 1000 = %1.3f',cost))
%
% Web Resources:
%   <a href="http://en.wikipedia.org/wiki/Dijkstra%27s_algorithm">Dijkstra's
Algorithm</a>
%   <a
href="http://en.wikipedia.org/wiki/Graph_%28mathematics%29">Graphs</a>
%   <a href="http://en.wikipedia.org/wiki/Adjacency_matrix">Adjacency
Matrix</a>
%
% See also: gplot, gplotd, gplotdc, distmat, ve2axy, axy2ve
%
% Author: Joseph Kirk
% Email: jdkirk630@gmail.com
% Release: 1.1
% Date: 4/29/09

% Errors corrected and function modified for use in SUPERMAN simulation by
% Darren Hurley-Smith, University of Greenwich, 2014

% Process Inputs
error(nargchk(2,5,nargin));
all_positive = 1;
[n,nc] = size(AorV);
[m,mc] = size(xyCorE);
[E,cost] = processInputs(AorV,xyCorE);
if nargin < 5
    iswaitbar = 0;
end
if nargin < 4
    FID = (1:n);
end
if nargin < 3
    SID = (1:n);
end
if max(SID) > n || min(SID) < 1
    eval(['help ' mfilename]);
    error('Invalid [SID] input. See help notes above.');
end
if max(FID) > n || min(FID) < 1
    eval(['help ' mfilename]);
    error('Invalid [FID] input. See help notes above.');
end

isreversed = 0;
```

```
if length(FID) < length(SID)
    E = E(:,[2 1]);
    cost = cost';
    tmp = SID;
    SID = FID;
    FID = tmp;
    isreversed = 1;
end

L = length(SID);
M = length(FID);
costs = zeros(L,M);
paths = num2cell(nan(L,M));

% Find the Minimum Costs and Paths using Dijkstra's Algorithm
if iswaitbar, wbh = waitbar(0,'Please Wait ... '); end
for k = 1:L
    % Initializations
    if all_positive, TBL = sparse(1,n); else TBL = NaN(1,n); end
    min_cost = Inf(1,n);
    settled = zeros(1,n);
    path = num2cell(nan(1,n));
    I = SID(k);
    min_cost(I) = 0;
    TBL(I) = 0;
    settled(I) = 1;
    path(I) = {I};

    while any(~settled(FID))
        % Update the Table
        TAB = TBL;
        if all_positive, TBL(I) = 0; else TBL(I) = NaN; end
        nids = find(E(:,1) == I);
        % Calculate the Costs to the Neighbor Points and Record Paths
        for kk = 1:length(nids)
            J = E(nids(kk),2);
            if ~settled(J)
                c = cost(I,J);
                if all_positive, empty = ~TAB(J); else empty =
isnan(TAB(J)); end
                if empty || (TAB(J) > (TAB(I) + c))
                    TBL(J) = TAB(I) + c;
                    if isreversed
                        path{J} = [J path{I}];
                    else
                        path{J} = [path{I} J];
                    end
                else
                    TBL(J) = TAB(J);
                end
            end
        end
```

```
        if all_positive, K = find(TBL); else K = find(~isnan(TBL)); end
        % Find the Minimum Value in the Table
        N = find(TBL(K) == min(TBL(K)));
        if isempty(N)
            break
        else
            % Settle the Minimum Value
            I = K(N(1));
            min_cost(I) = TBL(I);
            settled(I) = 1;
        end
    end
    % Store Costs and Paths
    costs(k,:) = min_cost(FID);
    paths(k,:) = path(FID);
    if iswaitbar, waitbar(k/L,wbh); end
end
if iswaitbar, close(wbh); end

if isreversed
    costs = costs';
    paths = paths';
end

if L == 1 && M == 1
    paths = paths{1};
end

% --------------------------------------------------------------------
    function [E,C] = processInputs(AorV,xyCorE)
        C = sparse(n,n);
        if n == nc
            if m == n
                if m == mc % Inputs: A,cost
                    A = AorV;
                    A = A - diag(diag(A));
                    C = xyCorE;
                    all_positive = all(C(logical(A)) > 0);
                    E = a2e(A);
                else % Inputs: A,xy
                    A = AorV;
                    A = A - diag(diag(A));
                    xy = xyCorE;
                    E = a2e(A);
                    D = ve2d(xy,E);
                    all_positive = all(D > 0);
                    for row = 1:length(D)
                        C(E(row,1),E(row,2)) = D(row);
                    end
                end
            else
                eval(['help ' mfilename]);
```

```
                error('Invalid [A,xy] or [A,cost] inputs. See help notes
above.');
            end
        else
            if mc == 2 % Inputs: V,E
                V = AorV;
                E = xyCorE;
                D = ve2d(V,E);
                all_positive = all(D > 0);
                for row = 1:m
                    C(E(row,1),E(row,2)) = D(row);
                end
            elseif mc == 3 % Inputs: V,E3
                E3 = xyCorE;
                all_positive = all(E3 > 0);
                E = E3(:,1:2);
                for row = 1:m
                    C(E3(row,1),E3(row,2)) = E3(row,3);
                end
            else
                eval(['help ' mfilename]);
                error('Invalid [V,E] inputs. See help notes above.');
            end
        end
    end

    % Convert Adjacency Matrix to Edge List
    function E = a2e(A)
        [I,J] = find(A);
        E = [I J];
    end

    % Compute Euclidean Distance for Edges
    function D = ve2d(V,E)
        VI = V(E(:,1),:);
        VJ = V(E(:,2),:);
        D = sqrt(sum((VI - VJ).^2,2));
    end
end
```

SUPERMAN_init_auth.m

```
% Function to calculate tne communication events and byte cost of
% SUPERMAN's in initialisation (network joining) phase
% Darren Hurley-Smith, University of Greenwich 2015

function [total_init, node, total_init_bytes] = SUPERMAN_Init_Auth(node,N)

DReq = 0;
CReq = 0;
```

```
CEx = 0;
SKb_send = 0;
netlist = [];
total_init = 0;
total_init_bytes = 0;

for n=1:N
    netlist(n) = node(n).netauth;
end

while sum(netlist) < N
    for n=1:N
        auth_flag = 0;
        if node(n).netauth ~= 1
            DReq = DReq+1;
            if auth_flag ~= 1
                for i=1:N
                    if i ~= n && length(node(n).rTable{i}) >= 1
                        this_route = node(n).rTable{i};
                        this_neighbour = this_route(2);
                        if node(this_neighbour).netauth == 1
                            node(n).netauth = 1;
                            netlist(n) = 1;
                            node(n).aTable(node(this_neighbour).Index) = 1;
                            node(node(this_neighbour).Index).aTable(n) = 1;
                            CReq = CReq + 1;
                            CEx = CEx + 1;
                            SKb_send = SKb_send + 1;
                            auth_flag = 1;
                            break;
                        end
                    end
                end
            end
        end
    end
end

total_init = DReq + CReq + CEx + SKb_send;
total_init_bytes = (DReq*(5+256+20)) + (CReq*(5+256+20)) +
(CEx*(25+1275+20)) + (SKb_send*(25+1275+256+20));


return;
```

SUPERMAN_init_auth.m

```
% Function to calculate tne communication events and byte cost of
% SUPERMAN's in initialisation (network joining) phase
% Darren Hurley-Smith, University of Greenwich 2015
```

```
function [total_init, node, total_init_bytes] = SUPERMAN_Init_Auth(node,N)

DReq = 0;
CReq = 0;
CEx = 0;
SKb_send = 0;
netlist = [];
total_init = 0;
total_init_bytes = 0;

for n=1:N
    netlist(n) = node(n).netauth;
end

while sum(netlist) < N
    for n=1:N
        auth_flag = 0;
        if node(n).netauth ~= 1
            DReq = DReq+1;
            if auth_flag ~= 1
                for i=1:N
                    if i ~= n && length(node(n).rTable{i}) >= 1
                        this_route = node(n).rTable{i};
                        this_neighbour = this_route(2);
                        if node(this_neighbour).netauth == 1
                            node(n).netauth = 1;
                            netlist(n) = 1;
                            node(n).aTable(node(this_neighbour).Index) = 1;
                            node(node(this_neighbour).Index).aTable(n) = 1;
                            CReq = CReq + 1;
                            CEx = CEx + 1;
                            SKb_send = SKb_send + 1;
                            auth_flag = 1;
                            break;
                        end
                    end
                end
            end
        end
    end
end

total_init = DReq + CReq + CEx + SKb_send;
total_init_bytes = (DReq*(5+256+20)) + (CReq*(5+256+20)) +
(CEx*(25+1275+20)) + (SKb_send*(25+1275+256+20));


return;
```

SUPERMAN_link_auth.m

```
% SUPERMAN Security Association script.
% Calculates the total number of communication events required to drive the
% network to a fully secured state, in which all nodes have secure links to
% each other
% Darren Hurley-Smith, University of Greenwich, 2015

function [total_comms, node, total_bytes] = SUPERMAN_Link_Auth(node, N)

auth_req = 0;
auth_rep = 0;
total_comms = 0;
total_bytes = 0;

for n=1:N
    for j=1:N
        if j~=n
            %check length of route between n and j for authenticated nodes
            if node(n).aTable(j) ~= 1
                this_route = node(n).rTable{j};
                for x=1:length(this_route)-1
                    if node(n).aTable(this_route(x+1)) == 1 &&
node(x+1).Index == this_route(2)
                        %send an auth request to this node, but do not yet
                        %reply (pass it down the route)
                        auth_req = auth_req + 1;
                    elseif node(n).aTable(this_route(x+1)) ~= 1 &&
node(this_route(x+1)).Index == this_route(2)
                        %send an auth request, authenticate with the node
and
                        %then repeat this process
                        auth_req = auth_req + 2;
                        auth_rep = auth_rep + 2;
                        node(n).aTable(this_route(x+1)) = 1;
                        node(x+1).aTable(this_route(1)) = 1;
                        x = x-1; %decrement counter to retry this node when
auth
                    elseif this_route(x+1) == node(j).Index
                        for z=length(this_route(x)):-1:1
                            if node(this_route(z)).Index ~= node(j).Index
                                if node(this_route(z)).aTable(j) ~= 1
                                    node(this_route(z)).aTable(j) = 1;
                                    auth_rep = auth_rep + 1;
                                else
                                    auth_rep = auth_rep + 1;
                                end
                            end
                        end
                    else
                        auth_req = auth_req + 1;
                    end
```

```
                end
            end
        end
    end
end

total_bytes = (auth_req*(25+20)) + (auth_rep*(25+256+20));
total_comms = (auth_req) + (auth_rep);

return;
```

IPsec_IKE_phase.m

```
%Darren Hurley-Smith, University of Greenwich 2015

%IPSEC IKE phase 1 negotiation phase simulation.  Provides session/mission
%net authentication to nodes

function [total_init, node, IPSEC_init_bytes] = IP_Init_Auth(node, N)

HELLO = 0;
CReq = 0;
CSend_Rem = 0;
Key_send = 0;
IPnetlist = [];
total_init = 0;
authflag = 0;

for n=1:N
    IPnetlist(n) = node(n).IPnetauth;
end

while sum(IPnetlist) < N
    for n=1:N
        auth_flag = 0;
        if node(n).IPnetauth ~= 1
            HELLO = HELLO+1;
            if auth_flag ~= 1
                    if 1 ~= n && length(node(n).rTable{1}) >= 2
                        this_route = node(n).rTable{1};
                        for j=1:length(this_route-1)
                            if this_route(j+1) == node(1).Index
                                node(n).IPnetauth = 1;
                                IPnetlist(n) = 1;
                                CReq = CReq + (length(this_route)-1);
                                CSend_Rem = CSend_Rem + (length(this_route)-
1);
                                Key_send = Key_send + (length(this_route)-
1);
                                break;
```

```
                            else
                                HELLO=HELLO+1;
                            end
                        end
                    end
                end
            end
        end
    end

total_init = (HELLO) + (CReq) + (CSend_Rem) + (Key_send);
IPSEC_init_bytes = (HELLO*(28+20)) + (CReq*(20+28+8+8+1+1+1+4+4)) +
(CSend_Rem*(20+28+8+8+1+1+1+4+4+657+32)) + (Key_send*(20+28+657+32));


return;
```

IPsec_Security_Association.m

```
% Darren Hurley-Smith, University of Greenwich 2015

% This function simulates the communication events (transmissions) sent
% when securing tunnels under IPSEC (MANET modified - MANIPSEC-like)

function [total_comms, node, total_bytes, total_init, total_init_bytes] =
IPSEC_Link_Auth(node, N)

auth_req = 0;
auth_rep = 0;
notify = 0;
total_comms = 0;

DReq = 0;
CReq = 0;
CEx = 0;
SKb_send = 0;
netlist = [];
total_init = 0;
total_init_bytes = 0;

for n=1:N
    for j=1:N
        if j~=n
            %check length of route between n and j for authenticated nodes
            if node(n).IPaTable(j) ~= 1
                this_route = node(n).rTable{j};
                DReq = DReq+((length(this_route)-1));
                for x=1:length(this_route)-1
                    if this_route(x+1) == node(j).Index
%                           node(this_route(x+1)).IPaTable(n) = 1;
```

```
                        node(n).IPaTable(j) = 1;
                        CEx = CEx + ((length(this_route)-1));
                        CReq = CReq + (length(this_route)-1);
                        auth_rep = auth_rep + ((length(this_route)-1));
                        auth_req = auth_req + (length(this_route)-1);
                        notify = notify + ((length(this_route)-1));
                    else
                        auth_req = auth_req + 1;
                    end
                end
            end
        end
    end
end

total_comms = (auth_req) + (auth_rep) + (notify);
total_bytes = (auth_req*(28+20)) + (auth_rep*(20+256+28)) + (notify*(20));

total_init = DReq + CReq + CEx;
total_init_bytes = (DReq*(28+20)) + (CReq*(20+28+8+8+1+1+1+4+4)) +
(CEx*(20+28+8+8+1+1+1+4+4+1275));

return;
```

SUPERAODV.m

```
% Function to calculate the communication events and byte cost of SUPERAODV
% Darren Hurley-Smith, University of Greenwich, 2015

function [bytes, node] = SUPERAODV(node, N)

bytes = 0;
RREQ = 0;
RREP = 0;
route = [];

for n=1:N
    for j=1:N
        route(n,j) = 0;
    end
end

    for n=1:N
        for j=1:N
            if n ~= j
                if route(n,j) == 0
                    if route(j,n) == 0
                        RREQ = RREQ + node(n).adjacent_nodes;
                        RREQ = RREQ + ((length(node(n).rTable{j})-1))-1;
                        RREP = RREP + ((length(node(n).rTable{j})-1));
```

```
                        for x=1:length(node(n).rTable{j})-1
                            RREQ = RREQ + node(n).adjacent_nodes;
                        end

                        route(n,j) = 1;
                        route(j,n) = 1;
                    end
                end
            end
        end
    end

bytes = (RREQ*102)+(RREP*98);
```

SUPEROLSR.m

```
% Function to calculate the communication events and byte cost associated
% with SOLSR
% Darren Hurley-Smith, University of Greenwich, 2015

function [bytes, node] = SUPEROLSR(node, N)

bytes = 0;
src = 0;


for n=1:N
        src = src + node(n).adjacent_nodes;
end


bytes = (src*94);
```

SAODV.m

```
% Function to calculate the communication events and byte cost associated
% with SAODV routing
% Darren Hurley-Smith, University of Greenwich, 2015

function [bytes, node] = SAODV(node, N)

bytes = 0;
RREQ = 0;
RREP = 0;
route = [];

for n=1:N
```

```
    for j=1:N
        route(n,j) = 0;
    end
end

    for n=1:N
        for j=1:N
            if n ~= j
                if route(n,j) == 0
                    if route(j,n) == 0
                        RREQ = RREQ + node(n).adjacent_nodes;
                        RREQ = RREQ + ((length(node(n).rTable{j})-1))-1;
                        RREP = RREP + ((length(node(n).rTable{j})-1));

                        for x=1:length(node(n).rTable{j})-1
                            RREQ = RREQ + node(n).adjacent_nodes;
                        end

                        route(n,j) = 1;
                        route(j,n) = 1;
                    end
                end
            end
        end
    end

bytes = (RREQ*125)+(RREP*121);
```

SOLSR.m

```
% Function to calculate the communication events and byte cost of SOLSR
% Darren Hurley-Smith, University of Greenwich, 2015

function [bytes, node] = SOLSR(node, N)

bytes = 0;
src = 0;
ack = 0;


for n=1:N
    src = src + node(n).adjacent_nodes;
    ack = ack + node(n).adjacent_nodes;
end


bytes = (src*44)+(ack*64);
```

AODV.m

```
% Function to determine the byte-cost and number of transmissions
% associated with AODV routing

function [bytes, node] = AODV(node, N)

bytes = 0;
RREQ = 0;
RREP = 0;
route = [];

for n=1:N
    for j=1:N
        route(n,j) = 0;
    end
end

    for n=1:N
        for j=1:N
            if n ~= j
                if route(n,j) == 0
                    if route(j,n) == 0
                        RREQ = RREQ + node(n).adjacent_nodes;
                        RREQ = RREQ + ((length(node(n).rTable{j})-1))-1;
                        RREP = RREP + ((length(node(n).rTable{j})-1));

                        for x=1:length(node(n).rTable{j})-1
                            RREQ = RREQ + node(n).adjacent_nodes;
                        end

                        route(n,j) = 1;
                        route(j,n) = 1;
                    end
                end
            end
        end
    end

bytes = (RREQ*77)+(RREP*73);
```

OLSR.m

```
% Function to calculate local flooding byte cost using OLSR for routing
% Darren Hurley-Smith, University of Greenwich, 2015

function [bytes, node] = OLSR(node, N)

bytes = 0;
src = 0;
```

```
for n=1:N
        src = src + node(n).adjacent_nodes;
end


bytes = (src*69);
```