# Netted Radar
# Modelling, Design and Optimisation

M Imran Khan
Cardiff University
Department of Computer Science

A thesis submitted in partial fulfilment of the
requirement for the degree of Doctor of Philosophy.

June 22, 2004





ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

DECLARATION

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed..M. Imran Hu.....(candidate)
Date...25/6/2004.....

STATEMENT 1

This thesis is the result of my own investigations, except where otherwise stated.

Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed...M. Imran Hu.....(candidate)
Date...25/6/2004.....

STATEMENT 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations

Signed..M. Imran Hu.....(candidate)
Date...25/6/2004.....

# ABSTRACT

Networks of phased array radars are able to provide better counter stealth target detection and classification. Each radar sensor/node generates information which requires transmission to a central control authority who is able to evaluate the information. This requires a communications network to be established to allow transmission of information to and from any node. Each radar node is limited by range and degree and relies on the formation of a multi-hop network to facilitate this transmission.

This thesis investigates a method whereby the radar beam itself is used in the formation of a multi-hop network. The phased array's multifunctional nature allows rapid switching between communications and radar function. A complete model of how the communication system could operate is included in this thesis. In order to simulate radar function and network communication, a custom built simulation platform was used.

Two different approaches to the formation of the radar networks are derived, one involving a global design method using the Strength Pareto Evolutionary Algorithm (SPEA) and another using a local method specifically tailored to radar network formation, Distributed Algorithm for Radar Topology Control (DARTC). Both of these techniques can be used in the formation of radar networks. The thesis goes further to suggest modifications to the model itself that could result in improved radar network communication performance.

# ACKNOWLEDGEMENTS

I would like to thank first and foremost my supervisor, Steve Hurley, who has had the patience and knowledge to guide me throughout this project. I must also thank Andrew Hume and Steve Harman of QinetiQ, who have provided valued feedback and background for my project. Thanks are also due to Cardiff University and QinetiQ for financially supporting my research.

I dedicate this thesis to my mother and father who have been truly supportive throughout my time at Cardiff University.

M. Imran Khan
February 2003

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ACRONYMS

| Acronym | Description |
|---|---|
| ADSL | Asymmetric Digital Subscriber Line |
| AOA | Angle of Arrival |
| ATM | Asynchronous Transfer Mode |
| CBDTC | Cone Based Distributed Topology Control |
| CDMA | Code Division Multiple Access |
| DARTC | Distributed Algorithm for Radar Topology control |
| FPAS | Fully Polynomial Approximation Scheme |
| GA | Genetic Algorithms |
| GBAD | Ground Based Air Defense |
| GPRS | General Packet Radio Service |
| GSM | Global System for Mobile Communication |
| LPF | Links per Face |
| MoD | Ministry of Defence |
| MOP | Multi-Objective Problem |
| MOGA | Multi-Objective Genetic Algorithm |
| MST | Minimum Spanning Tree |
| OS | Ordnance Survey |
| OSPF | Open Shortest Path First |
| RIP | Routing Information Protocol |
| RCS | Radar Cross Section |
| SA | Simulated Annealing |
| SD | Standard Deviation |
| TCP/IP | Transmission Control Protocol/ Internet Protocol |
| TDMA | Time Division Multiple Access |
| TS | Tabu Search |
| TSP | Travelling Salesman Problem |
| UMTS | Universal Mobile Telecommunications System |
| VT | Variable Timeslots |
| ZRP | Zone Routing Protocol |

CHAPTER

# ONE

# Introduction

The advantages and diversity of phased array radar have been known for decades. Fourikis [36] provides a rich overview and discusses the diverse range of applications of phased array based systems. More recent advances in phased arrays allow for inter-continental phased array communication. These radars communicate over vast distances and relay information to central control sites such as operation centres. This project uses much smaller radars to form tightly controlled radar networks over small geographical areas of up to 28x28 $km$ [80].

This thesis crosses the boundaries between radar and networking to produce radar networks allowing multiple target surveillance/tracking, multiple target perspectives and efficient message passing. It uses ideas from the wired network design community, wireless network design community and ad-hoc network community to develop a method for designing a multi-node radar network. These radars are relatively short range(7.5km [81]) and have characteristics that are similar to both wired and wireless networks.

The nature of the radar operation means that the radar network needs to be operational very fast in the order of 60 seconds i.e. the network design, link establishment, link synchronization need to be completed very quickly. When the radars are in position, the radar network needs to be operational in an efficient manner. The radars in this project are also considered to have a degree of mobility. Although they cannot communicate during movement, they can be repositioned. The design algorithms need to be very efficient and can therefore be considered rapid network design algorithms and with the mobility factor, they can be considered rapid network re-design algorithms.

Networks of radars of this type do not currently exist. The thesis is therefore not only concerned with the network design process but also how the complete com-

munication subsystem might operate. The radar model includes all communication aspects including protocol, timings, message design and routing. This model in turn influences which network design method could be used and how it could be applied to radar networks. It is therefore important that the model is an accurate representation of the actual radar communication subsystem in order that network designs can be evaluated.

The project does not make assumptions about operational control aspects of the system. A centralised control method would have different methods of controlling the network structure to a de-centralised approach. Currently, this question remains open not only to progress research into these types of radar network but also to analyse which is more appropriate, a centralised or decentralised approach.

This thesis:

1. builds a model of how radars could operate to form communication infrastructure.
2. develops appropriate rapid network design and redesign methods to form radar networks.
3. simulates the operation of radar networks to evaluate the performance of the radar network designs under simulation.
4. evaluates whether Radar networking as a concept is feasible.

This thesis does not:

1. attempt to find the best optimisation method available, only to decide if a particular type would be appropriate for use with radar networks.
2. analyse the true performance of network design algorithms applied to radar network design. This decision is based solely on whether or not the radar network performs adequately.

A conventional radar would see a stealthy target from a single perspective. A single radar would have only a single cross-sectional view of the target. By design this target will have a small cross sectional area from a single radar perspective. If this larger radar were replaced with 4 smaller radars figure 1.1 the radars would have different perspectives of the same target. It is also unlikely that the target will have a small cross section from any direction. Therefore 4 radars would have a better view of a stealthy target (see section 1.3.1 for a detailed description). The concept of improved stealthy target detection ability is the primary motivation for the development of a network of radars. Other motivations are given in section 1.2.3.

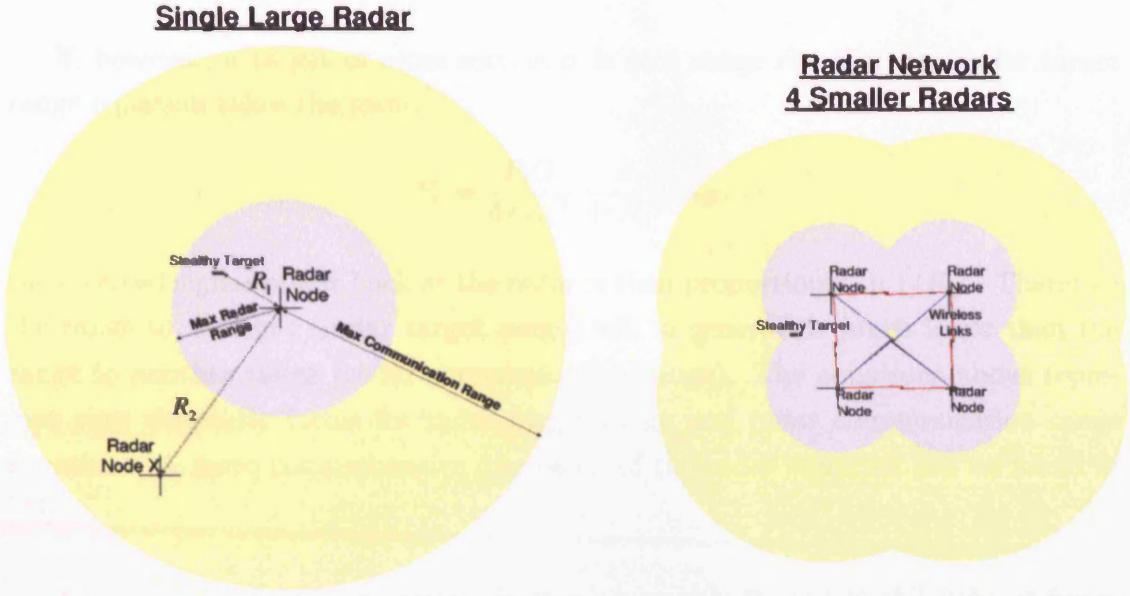



**Figure 1.1: Conventional Radar Operation (left) - Radar Network Concept (right) NB. Not drawn to scale**

In figure 1.1 above, the radar target range is shown as the grey area and the radar communication range is shown as the yellow area. Clearly the above diagram shows radar communication range is much larger than the radar target range. This can be explained using a simple communications range equation for a radar antenna. Consider the received power $P_r$ at radar node x at a range $R_2$ from the radar:

$$P_r = \frac{P_t G}{4\pi {R_2}^2}.A_E$$

where

$$\begin{aligned} \text{received signal power at receiver} &= P_r \\ \text{transmitter signal power} &= P_t \\ \text{antenna gain} &= G \\ \text{range from antenna} &= R \\ \text{effective antenna area} &= A_e \end{aligned}$$

Therefore another radar antenna at a distance $R_2$ will receive a power proportional to $1/{R_2}^2$.

If, however, a target of cross section $\sigma$ is at a range $R_1$, then the radar target range equation takes the form:

$$P_r = \frac{P_t G}{4\pi {R_1}^2} \cdot \frac{\sigma}{4\pi {R_1}^2} \cdot A_E$$

the received signal power back at the radar is then proportional to $1/{R_1}^4$. Therefore the range to a target (radar target range) will in general be much lower than the range to another radar (radar communication range). The equations above represent very simplistic forms for radar target range and radar communication range equations. A more comprehensive discussion of the radar equation can be found in [93].

The effect of the larger communication range is indicated in the right of figure 1.1. The radar target range does not overlap (grey circles do not cover the radar nodes). However the radar communication range does (yellow circles each cover all other nodes), allowing a communications network to be employed. In the example above, the data collected from all radars would then need to be aggregated at some central point (a Command Position) in order to determine a course of action. The radar network formed in Figure 1.1 above shows the network fully connected. This is acceptable for small networks but as the network sizes grow, both limited radar range and limited node degree require a multi-hop network to be designed. This network design and subsequent testing of these radar networks form the majority of this thesis.

## 1.1 Ground Based Air Defence Radar Networks

This project represents early research into communication of netted radars for Ground Based Air Defense (GBAD) using low power radar. Systems such as The Ericsson GIRAFFE system[1] is typical of what can be considered a network of radars. This system can extend coverage by adding more GIRAFFE systems which then communicate surveillance information back to a command and control site via a radio communication link. A separate subsystem is used to communicate this information. In contrast, this project attempts to use the radar system itself for inter-radar communication. Radar nodes in this project also have overlapping coverage which can give not only multiple perspectives but multiple perspectives at the same time. This results in multiple tracks for a single target which can be aggregated to provide better registration, identification and positional information [6].

[1] see www.ericsson.com/microwave/ for further details

Future GBAD radar is expected to take the form of a distributed architecture, separated spatially to maximise coverage and efficiency. These radars then need to communicate the information back to commander in a timely manner. These radars must utilise some form of communications network to relay this information.

## 1.2 Radar Networks

Networks of radars have the underlying problem of having significantly lower radar communication ranges due to the reduction in power as compared to the equivalent single larger radar (as indicated in figure 1.1). This prevents each node from communicating with all other nodes in a completely connected fashion. Therefore the radars must utilise some form of multi-hop communications network.

### 1.2.1 Integrating Communications Infrastructure

The obvious way of allowing these radars to communicate is to attach communications infrastructure such as radios to each radar node. Track information could then be sent via this attached communications infrastructure which acts as a communications relay. Integrating communications infrastructure would reduce information quality due to increased delays required to cross multiple communications platform (radar to radio) resulting in reduced tactical tempo. The cost of maintaining and supporting two communication systems might also be prohibitive.

### 1.2.2 Tactical Tempo

Tactical tempo is the overall process of network design, network redesign, network setup and network performance. Tactical tempo describes the ability of a system such as the radar network system to be setup, redeployed or repositioned, depending on battlefield requirements. Ryan [37] equates tactical tempo with flexibility. The system should be able to adapt quickly to any changes in the tactical environment and continue to provide communications for the intended application. The radar networks in this thesis are required to be located, designed, configured and ready for operation very quickly in order to support the GBAD role. If the network changes then tactical tempo must be maintained by adapting to the changes as quickly as possible.

### 1.2.3 Phased Array Radar for Communications

The phased array radar antenna has the ability to form beams in both transmit and receive, effectively allowing the radar to maximise its sensitivity in the desired direction while minimising its sensitivity in other directions. This versatility and agility can be used to good effect when trying to implement a communication function using the phased array radar itself. With this communications ability the phased array radar becomes a truly Multi-functional radar (MFR), performing both the GBAD and communications functions.

The use of Phased Array Radar Antenna for communication will allow several of the advantages associated with highly directional antennas as compared to their omni-directional conterparts. This results in an increased communications range which in turn allows more connections to be made. Directive antennas also have higher gains and therefore higher data capacity. All these factors contribute to having high data rate, low interference and well connected networks.

The collective use of a network of phased array antennas also has the advantage of being more difficult to jam because they have lower power than their larger equivalent. They also have lower sidelobes and ability to form adaptive nulls (place nulling beams in the direction of the jammer). These advantages are further enhanced because of the distributed nature of the radars, requiring jammers to interfere over a larger area and in multiple directions. Full details and analysis is given in [52].

### 1.2.4 Reconfigurable networks

In future battlefield scenarios where tactical tempo and manoeuvrability are key to the battle outcome, networks need to be reconfigurable and able to adapt. Nodes need to be placed based on tactical decisions dictated by the battle objectives and not communications. Commanders need freedom to deploy the network as required and a communications network needs to be established using methods that maintain communications quality without jeopardising tactical advantage.

As the node layout changes, whether through node movement, node destruction, network extension or node addition, the network should be able to rapidly re-configure itself to reflect the changes in the network and to maintain network quality.

This thesis attempts to facilitate the formation of radar networks that are able to perform as a GBAD system, without placing restrictions on the location of the radar nodes or commander.

## 1.3 The Military Environment

From a military perspective, being able to form a network of radars as set out in Figure 1.1 has a significant advantage over the single equivalent monostatic radar. A netted radar would allow multi-target perspectives, thus reducing the effect of stealthy targets. Other advantages include reduced vulnerability, reduced susceptibility to countermeasures and reduced obscuration effects [52].

### 1.3.1 Stealthy Targets

Difficult or low observable stealthy targets are likely to be the targets needed to be detected by future GBAD systems [81]. As stealthy targets fly towards a group of netted radars the stealthy nature of these targets is reduced. An example is shown in Figure 1.2. As the stealthy target approaches radar node A (point a), the effect of the stealthy designs' lower Radar Cross Section (RCS) reduces detection ability of node A. As the target enters node B's radar range (point b) the targets RCS increases as the target is no longer "nose on" to the radar. As the target enters node D's range (point c) a further perspective is gained for this target. The three separate tracks by nodes A,B and D help improve the detection ability, identification and tracking of the stealthy targets. As in figure 1.1, the radar range shown in grey in figure 1.2 reflects the radar target range.

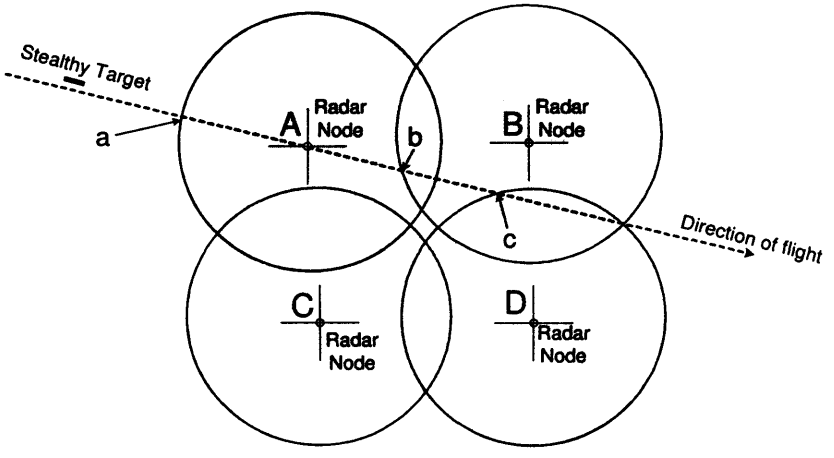



Figure 1.2: Radar detections for 4 netted radars

Full details of the concept can be found in [52]. In order to take advantage of this additional perspective information, a communications network needs to be utilised. Track information can then be sent for processing by a central control authority such as a commander located at any of the radar nodes.

### 1.3.2 Timeliness of Data

The function of GBAD systems is to protect military forces and infrastructure from enemy attack. The radar network should be able to provide information to an air defence commander in a timely and efficient manner. The air defence commander can then assess the information and determine a course of action. To enable the effective engagement of enemy targets, it is essential that the information is passed via a low latency network to give a clear indication of the current status of enemy targets. Therefore the prime measure for determining the effectiveness of the radar network is latency. This will be the measure of performance used throughout this project. Message delays of approximately 500ms or less are considered appropriate from an air defence commander's perspective [80].

## 1.4 Aims and Requirements

The primary aim of simulating a communications network is to determine how effective the system would be at passing information from the detection of targets, stealthy or otherwise. While the efficiency of this information passing is the key objective in radar network design, vulnerability of the network and hence reliability is also an important consideration in the formation of radar networks.

Work performed by QinetiQ has shown that there is a particular sensor/node density [2] that is most efficient at detecting a target with a given RCS profile [52]. This thesis attempts to enable the use of concepts discussed in [52] by providing a communications network for the radar network. This network can then be used to communicate track information back to commander located at any of the nodes in the network.

The thesis will examine the viability and feasibility of these GBAD netted radar networks in a simulated operational state. That is:

- Can these networks be built so that they can communicate efficiently?
- Can an adequate level of performance be achieved?
- Can tactical tempo be maintained?

In an attempt to answer some of these questions, the research focuses on three aspects in particular:

1. The Netted Radar Communication Model
2. Communication Network Design
3. Performance Evaluation

[2] Details of densities used within this project are given in later chapters

***The Netted Radar Model:*** The radar communication subsystem is modelled on predicted radar characteristics. The development of a complete radar communication model is fundamental to the project. Therefore the model itself represents a significant component of the research. Its also provides a benchmark for future work into this area of low power, short range netted radar networking.

***Communication Network Design:*** In order for the radars to pass messages, communications links will need to be established to form a connected network topology. A Network topology refers to the connection of nodes using communication links to form a single connected network. How and which communication links are used to establish a network topology are the central issues in network design. The issues involved in global (centralised) design and local (de-centralised) design are also considered from a performance and military perspective.

***Performance Evaluation:*** The network topology has to be evaluated in order to determine the quality in terms of message delays and network characteristics. Performance evaluation also incorporates other factors such as reliability and adaptability. However, performance of the network is primarily evaluated based on average message delay values to the commander. The quality of the track data and therefore the quality of the radar detection information itself, is not considered. Evaluating track data would involve analysing the messages passed to the command from a radar perspective (possibly including the detection signals themselves). A detailed discussion of the trade-offs in radar networking with respect to track data can be found in [6]. The decision as to whether the data is aggregated to form "measurement to track" (aggregating track information) or "track to track" (each track is considered independently) is considered beyond the scope of this study.

## 1.5 Radar Network Test System Overview

In order to fulfil the aims and requirements set out in the previous section, a complete evaluation system needs to be developed. The system has to incorporate the characteristics of the radar, its environment and the communications network. The system needs to be able to design the radar network and then simulate the radar operation in order to evaluate the overall system performance. A complete system outline is shown below in figure 1.3. This represents a complete overview of the expected form of a system to evaluate the new concept of netted radar communication.

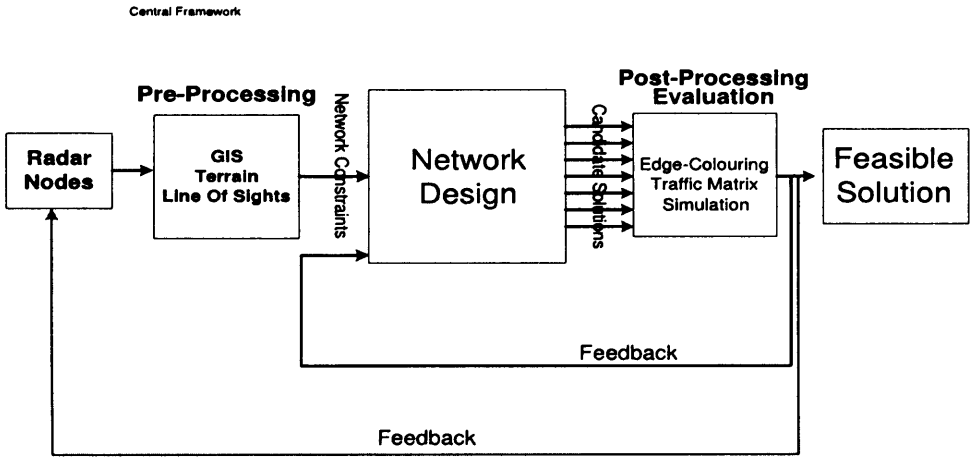



Figure 1.3: Radar network communication system and evaluation

***Radar Nodes.*** These are dual function nodes that carry out both the radar and communication functions. Each radar node has a target detection capability to perform the GBAD role, and communications capability to allow radar network formation. The radar detection and communication characteristics are outlined fully in section 3.

***Pre-Processing.*** The pre-processing is done mainly by a Geographical Information System (GIS) to generate line of sight and positional information from real terrain.

***Network Design*** The nodes must then be connected via communication links to form a communications network. Where a link is the formation of a wireless directive beam from one radar node to another. Which links are established are decided by the network design algorithms.

***Post-Processing and Evaluation*** The networks are post-processed and the radars are set-up to ensure that the correct links and timeslots have been allocated

(see section 3). The purpose of the evaluation system is to generate network traffic in response to a series of targets and then to analyse the performance of the radar network design with respect to message delays, routing, and network characteristics. The key performance measure is average message delay time to the commander. Also, information gained at this point may enable correction or improvement in the network design phase of the system through feedback.

***Feedback*** This is an useful aspect of this project. The entire system is being designed based on a predicted radar model. The current state-of-the-art radars are not currently able to perform as described in this thesis. The model is based on predicted radar performance[82]. Therefore this thesis provides not only feedback into the design process but also into possible development of the actual hardware (red feedback loop in figure 1.3).

***Feasible Solutions*** After the network design has been evaluated, a feasible solution can be determined if it has performed adequately.

## 1.6 The OSI model

The OSI 7 layer model [108] is shown in figure 1.4 below. It describes the different logical aspects of open systems communication interconnection. A complete overview of the OSI model can be found in [96].

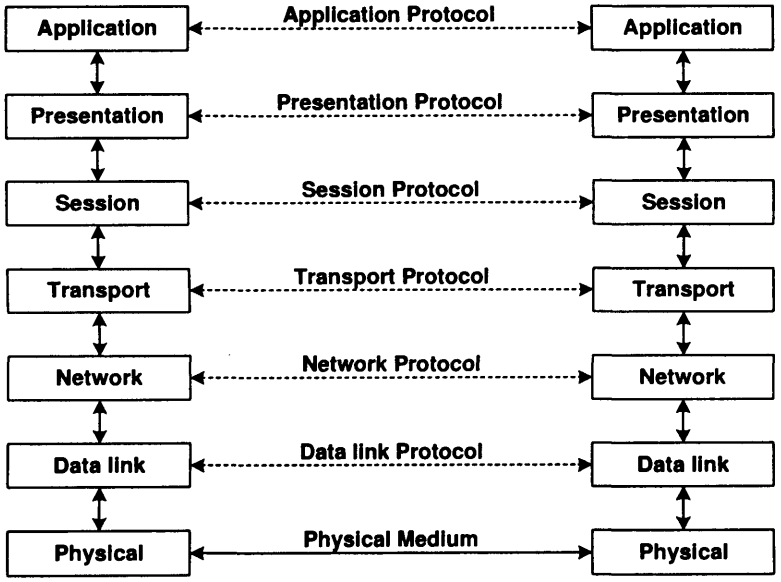



Figure 1.4: OSI 7 Layer reference model

This thesis looks at radar networking concept at three levels, the physical, data-link and network layer. The primary aim is to determine how effective a radar network would be at delivering messages from a source to a destination. What is done with these messages after they arrive at their desitiation is not considered. Therefore, from the transport layer up is not considered. To that end, various aspects of the physical layer are investigated such as how the radar controls the timeslots and how the radars interoperate with each other(section 3.2). The data link layer protocol is looked at superficially as additional fields in a radar message (section 3.2.3). Network layer protocols are considered to allow the radar network to deliver packets from source to destination in an effective manner.

## 1.7 Thesis Outline

This thesis contains 12 chapters, an outline of which is given below:

***Chapter 2*** outlines the network design concept. It also gives an overview of traditional and modern heuristic methods of network design. Optimisation techniques are discussed in terms of single and multiple objective approaches. The chapter also puts the radar network concept as a network design problem into context.

***Chapter 3*** describes the model developed for netted radar networking. It describes the radar characteristics, communication operation and event generation. A simulation environment (developed in this thesis) for netted radars is also described.

***Chapter 4*** shows how radar networks can be designed using a global perspective. It also describes designing radar networks using the multi-objective technique SPEA. The algorithm performance as a radar network design tool is also analysed.

***Chapter 5*** presents an analysis of the globally designed radar networks in terms of simulated performance. It gives an indication of algorithm performance inferred by simulation. The number of links per face is also analysed.

***Chapter 6*** introduces a local radar network design method specifically developed for this application, DARTC. Its performance is analysed and then compared to the global design equivalent. Local versus global design methods are also discussed.

***Chapter 7*** develops an idea to overcome some of the reliability issues involved in producing radar networks with 1 link per face. The design methods and implementation issues are also discussed using worked examples.

***Chapter 8*** looks at how node movements are handled. Node additions are considered in terms of multiple node additions to extend radar coverage and single node additions at arbitrary positions throughout the network.

***Chapter 9*** suggests an alteration that can be made to the radar communication system that can enhance performance significantly. Variable timeslots are introduced and their impact on performance is analysed. The performance increase is also compared when used in conjunction with the global SPEA and local DARTC design methods.

***Chapter 10*** uses the design methods developed and applies them to gradually increasing network sizes. The design methods are evaluated both in terms of algorithm scalability and radar network performance scalability.

***Chapter 11*** looks at the impact terrain has on netted radar design. Two examples using different terrain are analysed.

***Chapter 12*** draws conclusions from the thesis and the netted radar concept developed. It also suggests further work and research into the netted radar concept.

CHAPTER

# TWO

# Network Design

Central to this thesis is the concept of network design. The radars must form a communications network in order to process target information from the radars (see figure 2.1). The construction of this network is fundamental to this project. The following section introduces the concept of a network and network design methods.

## 2.1 What is a Network?

A graph $G$ is defined by its vertex set $V$ and its edge set $E$. Each vertex in the vertex set is called a node or resource in network terms. Each edge in the edge set is known as a link or communications link in network terms. An edge consists of a line or wireless connection between nodes $u_i$ and $u_j$, where $u_i$ and $u_j$ are both vertices in $V$, therefore an edge $e = \{u_i, u_j\} \in E$. A communications network can then be represented by a graph where $G = (V, E)$.

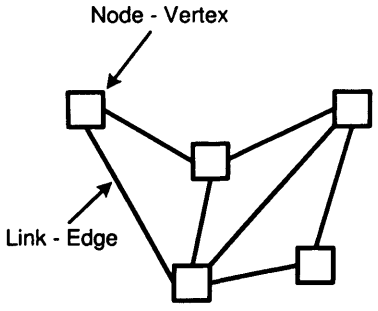



Figure 2.1: Network Graph

### 2.1.1 What is Network Design?

Network design is creating a structure by which a series of nodes can exchange information and operate as intended. Decisions have to be made based on how to allocate and connect resources. The question that is important to network design in radar networks is:

- How can a radar network be designed in order to provide the best performance (least delay) without knowing the position of the commander node?

In conventional fixed line network design, the cost(price) of the link generally dictates the possible data rates achievable. For example, expensive fibre optic cables have higher data rates than cheaper copper cables. In addition to being able to control the link capacity, the position of the server nodes or nodes that have particularly high traffic requirements is usually known. Networks can then be designed based around these particular nodes.

In radar networks, the link capacities are determined by the characteristics of the nodes(power, gain, etc) and the distance between the radar nodes. Radar networks have the added difficulty of not being able to determine before network design where the commander is located and therefore where the higher volumes of traffic will be destined. In order to design well performing radar network design, the following have to be considered:

1. A suitable combination of network design objectives that allow for very fast network design and re-design.
2. Design objectives that are indicative of well performing radar networks.
3. Design objectives that contribute to good network design. These objectives could also be conflicting and a required balance/tradeoff between them might be required.

The procedure is further complicated because the network design needs to take into account other factors such as network information flows between nodes, maximum flow, reliability, etc. Designing such networks can be an *NP*-complete problem (See Tim [13] for a discussion of the implications of *NP*-completeness on network design).

**Definition** *NP*-complete problems are both *NP* (no known polynomial time algorithm is able to solve the problem) and *NP* hard (any other *NP*-problem can be translated into this problem).

Algorithms that attempt to form solutions to *NP*-complete problems have been developed to give good solutions in a reasonable time rather than finding the optimal solution in an non-deterministic time.

Radar network design is constrained by node location which directly affects the possible node connections and capacity of the links. Radar network design is further constrained by having both a limited number of links per node and per sector. This is defined in section 3. These constraints must be taken into account during the network design process.

### 2.1.2 Network Criteria

The algorithms or heuristics used to design topologies use different network criteria depending on the intended application and design goals of the network. These can include:

- ***Maximum Flow:*** Maximise flow between a single source and single sink. This measure attempts to determine the capacity required to route the required flow between a pairs of nodes efficiently. Maximising the flow allows the maximum amount of data to be transferred from source to sink utilising the highest capacity routes. This increased capacity leads to more efficient message passing through higher throughput and lower delays. Maximum flow can be written as shown below where the total flow $f$ is the sum of all flows between a distinct pair of nodes $s$ and $t$.

$$\sum_{s,t \in V} f(s,t)$$

  Maximum flow problems can be solved using well-known and efficient algorithms such as the Ford-Fulkerson maximum flow algorithm. Maximum flow algorithms can also be used to determine different aspects of reliability.

- ***Multi-Commodity Flow:*** The multi commodity flow problem is essentially multiple maximum flow problems where there are multiple sources and sinks rather than one of each. For example, a telecommunications network where several messages must be transmitted between multiple nodes simultaneously is a multi-commodity flow problem. Being able to improve this multi-commodity flow problem would allow messages to be passed to their destination sinks more efficiently. As the routing problem for a network consists of finding paths or multiple paths from different sources and sinks, it is natural to model the traffic/message transfer problem as a multi-commodity network flow problem. Examples of the routing problem as a multi-commodity flow problem can be found in [70].

The total flow between the sets of source and sink X, Y vertices is defined as:

$$f(X,Y) = \sum_{x \in X} \sum_{y \in Y} f(x,y)$$

Whilst this problem can be reduced to a series of maximum flow problems, the execution times for multi commodity flow calculations can be significant. Using multi-commodity flows as criteria for network design can therefore be extremely computationally expensive. (Cormen [16] and Cook [89] give a complete account of the multi-commodity flow problem and its difficulties in combinatorial optimisation). A secondary problem with the multi-commodity flow calculations is the accurate traffic prediction models that are required to give useful results. [57], [72].

- ***Delay:*** Another method of measuring the quality of a network is to estimate the delay involved in transmitting messages in a network. This involves probabilistic methods of estimating message distributions, arrivals, and queuing [59]. Again the accurate prediction models are important in gaining useful results.

- ***Routing:*** A topology can be analysed based on how efficiently messages can be routed though the network. This can be estimated based on topology characteristics. An example is the number of hops along the shortest paths from all sources and sinks using the all-pairs shortest paths algorithms (see section 4.2.1 for an example. Alternative methods are given in [16]). Other methods include explicit routing where source and sink pairs reserve bandwidth along pre-determined explicit routes in order to guarantee bandwidth is available. Examples of these routing metrics are given in [57].

  The more complicated the routing strategy, the greater the algorithm complexity and time required to evaluate the network. Simple routing methods can be evaluated based on topology design alone. Routing algorithms that have to be analysed using traffic prediction models or traffic requirement models take considerably longer to evaluate their performance.

- ***Reliability:*** A topology design goal can be creating a reliable network for minimum cost. This means designing networks that have some element of redundancy in the event of failure. This can be as simple as minimising the networks edge or vertex connectivity [105] between a single pair of nodes, to complex measures of reliability such as toughness [14]. Simple edge or vertex connectivity can be solved in linear time. However, measures such as toughness are themselves *NP*-complete problems and therefore incorporating them

into network design heuristics is extremely computationally prohibitive. Reliability is discussed in later chapters.

- ***Cost:*** The cost(price) of the network can be calculated based on the type of line, its capacity, link length etc. There can also be other factors such as fixed costs and installation costs. Finding the optimal configuration for a network topology based on a set of discrete link sizes involves searching a large number of solutions. For example if there are $n$ nodes and $p$ discrete link sizes between these nodes, then evaluating all possible solutions would involve a search space of $p^{\frac{n(n-1)}{2}}$.

## 2.2 Heuristic Network Design

**Definition** A heuristic is a rule of thumb used to help solve problems where no exact procedure for doing so exists [73]. Heuristic network design incorporates design principles and applies them like rules of thumb in an attempt to produce a good network design.

Most of the network design problems involve search spaces that are extremely large. Therefore heuristic methods are used to provide sub-optimal designs in a reasonable time[1]. Heuristics used in fixed line topology networks can be classified into two groups, conventional or classical heuristics and evolutionary or modern heuristics. It is important to note that although modern heuristics are developed using new and innovative methods, their performance is not necessarily going to be better than their classical counterpart. If detailed information regarding the network is available, the network designer might be able to build a "Classic" network design algorithm that uses problem specific knowledge that produces better results than any modern technique.

### 2.2.1 Conventional Heuristics

There are many algorithms that attempt to solve the network design problem in a set deterministic or greedy approach. Some examples are:

- ***Minimum Spanning Tree:*** This is the simplest of all network design heuristics as it guarantees the cheapest or shortest subset of edges that connects all the nodes. The minimum spanning tree (MST) can also be calculated very quickly using Kruskal [61] or Prim algorithms [79]. Constructing minimum spanning trees suffers from scaling problems as the number of nodes increases. Complete analysis of the algorithms can be found in [16]

[1] A reasonable time is determined by the nature of the problem

- ***Esau-Williams:*** The Esau-Willaims [30] method attempts to overcome some of the deficiencies associated with the MST design process. The algorithm bases decisions on the capacity of links to avoid overloading central nodes and links. It essentially reduces the average number of hops.

- ***Tours:*** Tours introduce an element of reliability by making design decisions in order to guarantee that the graph remains connected in the event of single node failures. Tours are constructed in a similar manner to that of the travelling salesman problem TSP. The tours algorithm also suffers from scaling problems as the number of nodes increases. ( Cahn [9] gives a detailed account of the tour design process).

- ***Mentor:*** The Mentor algorithm [38] is another greedy algorithm that bases design decisions on how the network nodes are partitioned into concentrator or end nodes. The Mentor algorithm uses aspects of both the Esau-WIlliams and MST methods.

These algorithms are general network design algorithms. There are countless problem specific network design algorithms.

### 2.2.2 Modern Heuristics

Rather than having a deterministic approach to finding a good solution, modern heuristics attempt to increase gradually the quality of the solution. This involves evaluating a fitness function $f(x)$ at each stage of the algorithm. The fitness function could be any network criteria such as those included in section 2.1.2. The following methods are examples of modern heuristic techniques:

- ***Local Search:*** This is a simple adaptation to exhaustive search. A solution is picked at random from the search space. This solution is then evaluated based on some fitness function $f(x)$. A transformation T is then applied to the current solution and a new solution generated. If the new solution is better, the current solution is replaced with this new solution. These transformations are then applied until no further improvements can be made. A different starting point is then chosen and the complete algorithm is repeated. This is an example of a local search hill-climbing algorithm that uses an iterative approach to obtain a better solution.

- ***Simulated Annealing:*** Simulated Annealing (SA) [58] is very similar to local search except that a new solution can be accepted with a probability $p$ even if it is worse than the current solution.

Simulated annealing gradually reduces a parameter $T$ so that the algorithm gradually focuses on better solutions. This prevents the algorithm from getting stuck permanently at local optima. Take, for example, an initial starting solution $s'$ and solution $s''$ within the neighborhood. A random number, r, is then generated between 0 and 1. If the probability function $e^{(f(s'')-f(s'))/T} > r$ then the next solution is accepted. The temperature $T$ is then decreased until the algorithm becomes a conventional hill-climbing local search algorithm.

- ***Tabu Search:*** Tabu Search (TS) [40] is a local search method. Tabu search characterizes a transformation of one solution to another as a "move". The main difference between TS and other local search methods is that TS uses "memory" structures. These memory structures cause the algorithm to explore new areas of the solution space by making previous solutions and moves forbidden for a specified time. The complete details of the algorithm can be obtained in [40]. (Xu [107] gives a detailed account of using tabu search with the network design problem).

- ***Genetic Algorithms:*** Genetic algorithms (GA) are not local search methods. They are described as evolutionary algorithms as they operate in a way that can be described in a biological analogy. For instance, genetic algorithms use terms such as mutation and crossover to describe methods/operations that occur to solutions during the algorithm. Their operation, whilst seemingly straightforward, is difficult to analyse from a mathematical perspective. An excellent overview of the theory and background is available in [66]. Genetic Algorithms have been successfully applied to network design problem. [1] [2] [22] are just a few of the GA approaches successfully applied to the network design problem.

  Networks can be easily encoded into a chromosome consisting of $\frac{n(n-1)}{2}$ bits, where each bit represents a network connection. Each bit then corresponds to the existence of a link between a pair of its $n$ nodes. A variable length integer string representation can also be used to encode a network whereby every possible arc is given an integer value. The presence of a link can be indicated by its corresponding integer value in the string [22]. Other more novel methods of network encoding such as Node-Pair encoding are described in [28]. Each have advantages and disadvantages in terms of memory requirements, operator complexity and implementation complexity.

Modern heuristics can be classed as traditional or evolutionary. Both methods attempt to explore the search space in a structured manner. A comprehensive introduction to modern heuristic methods can be found in [19].

### 2.2.3 Wireless Networks

Commercial wireless networks require dedicated fixed base stations to service wireless clients. Technologies such as GSM [67], GPRS [39], TDMA [2] and CDMA [3] [41] all rely on base stations to provide coverage for users in a cellular environment. Indeed the positioning of these base stations is a complete optimisation problem in itself. More recently the IEEE 802.11 [42] protocol has widely been used in both commercial and home based applications for providing wireless networking. Cellular networks are design based around the one-hop principle, ie. one hop to wired central infrastructure. The 802.11 protocol does offer some multi-hop (Peer-to-Peer) support but over very short ranges and a very limited number of users.

In tactical environments, mobility of the entire infrastructure is required. This requires movement of central infrastructure such as base stations. In a military environment, these tactical networks are required in order to operate in any geographical location and provide high levels of service. Systems such as the Harris Falcon [4] which are US military based tactical and data networking radios, use this dynamic base station approach. The system employs a series of access points to service a set of users. These access points themselves are wireless, thus forming a completely dynamic network. This system is similar to how Bowman[5] radios will operate with future field MoD voice and data communications. This hierarchical deployment allows the entire system to be highly scalable, i.e. increase the number of access points as user numbers or demands increase.

Wireless networks that have no central infrastructure form the third class of wireless networks, Ad-hoc networks. Ad-hoc networks [76] are networks where the network users and clients form networks in an ad-hoc fashion. Each node/client makes decisions that contribute to the overall effect of producing a connected network. Each node must then operate as a forwarding routing node. It is therefore important that the correct links are chosen between nodes. Routing[76] also plays an important role in load balancing and ensuring routing remains effective in a dynamic environment.

[2]Time Division Multiple Access is a communications technique that uses a common channel(frequency) for communications among multiple users by allocating unique time slots to different users

[3]Code-Division Multiple Access, unlike TDMA, does not assign a specific frequency to each user. Instead, every channel uses the full available spectrum. Individual conversations are encoded with a pseudo-random digital sequence.

[4]http://www.rfcomm.harris.com/products/tactical-networking-data/

[5]http://www.generaldynamics.uk.com/

Wireless networks can therefore be divided into 3 groups:

1. Fixed backbone or fixed base station - Wireless devices are serviced by high power fixed location base stations which are connected typically by high capacity fixed cable/fibre optic links.
2. Dynamic base station, dynamic infrastructure - Wireless devices are serviced by mobile base stations. Dynamic base stations communicate with other dynamic base stations using higher power wireless links. The structure is inherently hierarchical. An example of such a system is the Bowman[6] system.
3. Instant infrastructure ad-hoc networks - Each wireless device allows other devices to be connected to it in a peer-to-peer manner and rely on each other to route information and ensure the network remains connected.

### 2.2.4 Wireless Network Design

Wireless networks have different requirements to wired networks with the obvious difference in transmission medium. Traditional fixed base station wireless networks are designed based on a number of issues including users, service level requirements, frequency assignment [29] considerations and interference. This thesis does not use fixed infrastructure although frequency constraint aspects will be considered later. Designing fixed base station networks is essentially comprised of two distinct design processes. The first process involves base station placement and optimisation like those described in [106] and [10]. The second is the optimisation based on users, requirements, service levels, frequency assignment [29]. Depending on the network type the second process is subdivided. TDMA 2G systems network planning is usually subdivided into a coverage problem [64] and a frequency assignment problem [29]. Newer technologies such as UMTS[7]require slightly different optimisation methods which are derived solely from coverage such as those described in [3].

Both dynamic base station and ad-hoc networks primarily focus on energy deficiency as the main criteria for network design. ( Milner [99] discusses the design of tactical networks with limited power the central focus of the investigations). Energy efficiency design is discussed in [101] for traditional non-military ad-hoc networks. Routing implementations for dynamic ad-hoc networks have also been widely researched. Early routing algorithms and implementations such as Mobile IP [75] have been superseded and extended with more advanced algorithms such as AODV [74]and DSR [55]. A comprehensive overview of the issues involved in mobile ad-hoc networking can be found in [45] and [76].

[6]http://www.generaldynamics.uk.com/

[7](UMTS) Wideband Code-Division Multiple-Access technology works by digitizing and transmitting the signals in a coded, wide spread spectrum over a range of frequencies.

### 2.2.5 Radar Networks and Radar Network Design

The radar network design concept as discussed in this thesis is unique. No current communication protocol is able to support radar networks. The timeslots method (see section 3) of communication would require a new protocol to be implemented to provide control. The timesharing requirement of the protocol is similar to TDMA in that it shares the communication channel by allocating timeslots to different nodes. The new protocol would be required to maintain the radar network timings, communication and medium access.

Studies such as [63] consider network topology management and design for large phased array antennas with long ranges and high data rates. The formation of these networks is also based on a straightforward network mesh formation as the large ranges enable easy link establishment between any nodes. These phased arrays are also considered to be used solely as high bandwidth backbone nodes with communication the priority.

Radar Phased array radar networks considered here have characteristics that collectively make their use innovative:

1. *Their primary function remains Ground Based Air Defence, not communication.* The phased array antenna's main concern is to be able to detect targets using scanning techniques associated with this type of antenna. The phased array radar can also make use of the highly directional links to form communication links with other networks. This communication function cannot be allowed to reduce the radar function capabilities of the radar. For this reason the radar performs the radar function for 90% of the available time and the remaining 10% is used to perform communications.

2. *A multihop network is built to form a communications network.* The concept outlined in section 1.1 discussed the method of replacing a single large radar with a set of smaller lower powered, lower range radars. This reduction in power prevents each radar communicating with any other radar directly. The radars must utilise multi-hop routes to enable communications with any other radar in the network. This use of phased array radars to form a connected network topology is new to GBAD.

3. *The communications network is used to send target track information and maintain tactical tempo.* Messages produced by the radars are sent via the established radar network to a commander node. This method of GBAD requires that these messages be formed and sent via the radar network. The message creation and message passing though the radar network is also new to GBAD.

From a network design perspective, the nodes can be considered ad-hoc nodes but the decision as to the nature of the network design has not been established and is a central focus of this project. Radar networking design using genetic algorithms and multiobjective techniques is described in the following sections.

### 2.2.6 Genetic Algorithms and Network Design

The Genetic algorithm technique was invented by Holland [48]. Davis [20] describes a genetic algorithm as having 5 components. These components are discussed with respect to network design:

1. ***A method of encoding the problem in the form of a chromosome.*** Encoding radar networks has already been discussed briefly. In this thesis the binary encoding method is used and is described in detail in section 4.2.

2. ***A function that evaluates the "fitness" of the solution.*** The fitness function in network design can consist of evaluation metrics discussed in section 2.1.2. A single or combination of network metrics can be used in a fitness function.

3. ***A method of obtaining an initial population of solutions.*** Initial network designs can comprise of a fully connected network, empty network or randomly generated string. The method is problem specific.

4. ***Reproduction operators for encoded solutions.*** Conventional crossover and mutation can be applied to binary encoded strings whilst other network representations require custom crossover and mutation depending on the representation.

5. ***Appropriate settings for the genetic algorithm control parameters*** These settings include population size, total number of generations, crossover rates, mutation rates. Different network design problems require the appropriate problem specific parameters.

These components allow the search process to be a computationally simple process and yet have powerful search capabilities without having any dependence on domain knowledge (knowledge about the problem space) or the local space around the solutions.

Given the componenets just described, the search process proceeds as shown below:

1. Create an initial population of chromosomes.
2. Evaluate the fitness of each chromosome.
3. While the population shows sufficient diversity-
   (a) Select a pair of chromosomes using some random selection procedure.
   (b) Perform crossover to allow chromosomes to exchange information.
   (c) Apply mutation with some probability dictated by the application.
   (d) Evaluate the fitness of new offspring.
   (e) Replace some or all of the previous population with the offspring population.

The process repeats until it is terminated when there is no longer any further useful information being exchanged and the GA converges on a final solution.

The natural translation of a network topology into a binary string make the operation of genetic algorithm straightforward. Methods of network design using genetic algorithms have been successfully implemented using a number of different network encodings [28]. Both Kershenbaum [71] and Michalewicz [65] attempt to explain why genetic algorithms maintain good performance with the network design problem using schema theorems. The interested reader should refer to [71] for a full analysis of the genetic algorithm approach to network design.

With very little domain knowledge about the radar network design problem, GA's seemed immediately obvious to use in conjunction with radar networks. This, along with the GA's proven performance with the network design problem, made their use with the radar network design appropriate. GA network encoding remains the same throughout this thesis to keep results consistent. As outlined in section 1, it does not attempt to find or evaluate the different methods of encoding and optimisation, but only to facilitate the optimisation of radar network design.

### 2.2.7 Single Objective Network Design - Weighted Method

All approaches to network design methods described in section 2.2 use criteria such as those described in section 2.1.2 and fitness functions to drive the algorithms or evaluate the final result. However, many network designs require simultaneously trading off one criterion for another. For example, attempting to minimise the price of the network while attempting to minimise delay. These two criteria are conflicting as decreasing price will tend to increase delay. This results in a number of solutions with different prices and different delays. This forms a multi-objective problem (MOP).

The original MOP technique as described in [15] forms a linear equation consisting of individual components, $f_i$:

$$\text{maximise} = f(x) = w_1.f_1(x) + w_2.f_2(x) + ..... + w_k.f_k(x)$$

The fitness function $f(x)$ is determined by the weights. These weights are then varied according to the relative merits and required pressure to guide the search towards required solutions.

#### Simple Multi-objective Network Design Problem

The following is an example of a simple network function with three criteria as shown below:

$$\begin{aligned} f(x) &= w_1.f_1(x) + w_2.f_2(x) + w_3.f_3(x) \\ \textit{where} \quad f_1 &= \text{Average Node Degree} \\ f_2 &= \text{Average Number of Hops} \\ f_3 &= \text{Average Link Length} \end{aligned} \tag{2.1}$$

A very simple GA that searches for solutions based on $f(x)$ is run. The weights are then varied according to the relative merits and required selection pressure to guide the search towards required solutions.

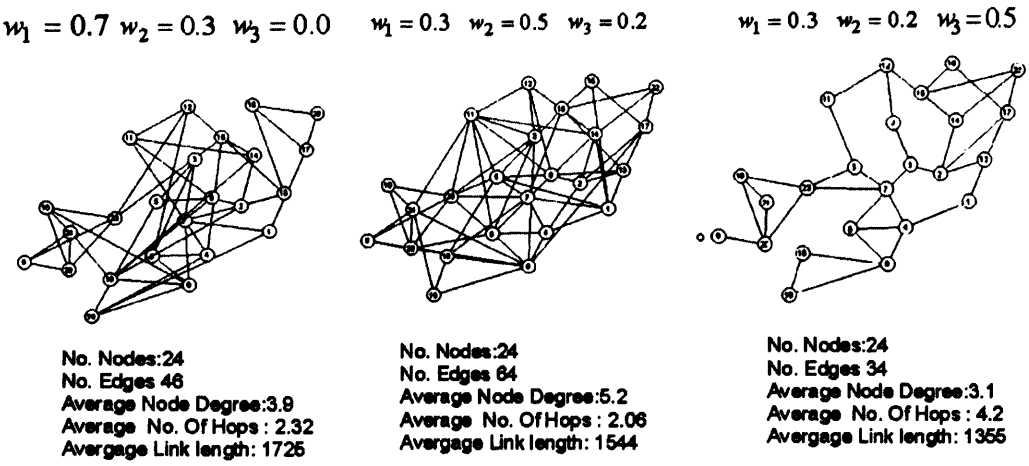



Figure 2.2: Weighted method optimisation

Figure 2.2 shows how varying the emphasis on the weights causes the characteristics of the networks to change. In the leftmost figure, the weights are biased towards node degree and because of this it has the best value i.e. approaching 4 links per node for this example. The centre figure is biased towards number of hops as shown by the increase in the number of links and reduced average number of hops. The rightmost figure is biased towards link length as shown by the reduction in the number of links and removal of the longer links.

### 2.2.8 Problems of Weighted Methods

It is clear that although being able to form a simple $f(x)$ fitness function will make them applicable to many optimisation techniques, there are a few serious drawbacks:

1. ***They require good domain knowledge.*** In the previous example it would be difficult to vary the weights effectively unless there is knowledge about what the effects would be.

2. ***Different weights require different runs.*** For each set of weights applied to the function, the optimisation has to be run. Therefore changing the weights will require additional runs. This raises two further issues, firstly what ranges of weights to use and secondly how many weight changes to test. Running additional optimisations also requires additional time.

For an in-depth discussion of linear functions in multi-objective design see [21].

## 2.3 Multi-objective Optimisation: An Overview

The simultaneous optimisation of a problem with multiple conflicting criteria to form a set of solutions characterises multi-objective design [33]. The ability to form this set of so called Pareto optimal solutions without the interaction of an outside influence or Decision maker (DM) make multi-objective algorithms important in optimisation. For a comprehensive overview of the various multi-objective optimisation techniques see, [33] and [21].

### 2.3.1 Pareto Optimality

Multi-objective optimisation uses the idea of a pareto front to describe a set of solutions that are equally good. Consider the example shown in figure 2.4 where $f1$ is a performance objective and $f2$ is a cheapness objective (inverse of price). With respect to figure 2.4:

- Solution B is superior to solution C as it has a higher performance and is cheaper. Solution C, although as cheap as D, is higher performing and is said to weakly dominate D.
- Solution E has a higher performance than B but is not as cheap. B is cheaper than E but has lower performance. Therefore E and B are indifferent.
- A is cheaper and has higher performance than B. Therefore A outperforms B in all respects.

The following three relations define the above characteristics where $a$ and $b$ are the decision vectors:

$$\begin{aligned} a \succ b \ (a \text{ dominates } b) \quad & \textit{iff} \quad f(a) > f(b) \\ a \succeq b \ (a \text{ weakly dominates } b) \quad & \textit{iff} \quad f(a) \geq f(b) \\ a \sim b \ (a \text{ is indifferent to } b) \quad & \textit{iff} \quad f(a) \not\geq f(b) \wedge f(b) \not\geq f(a) \end{aligned}$$

Now consider figure 2.4 with respect to solution B. We can now say that B *dominates* C and D and *is dominated* by solution A. Solution E is however, indifferent to B in that, although it is not as cheap, it has higher performance. Solution E is therefore neither worse nor better than solution B. Multi-objective optimisation techniques use pareto dominance as just described to evaluate solutions and form a set of non-dominated solutions, as shown in figure 2.3, known as the pareto front or pareto set.

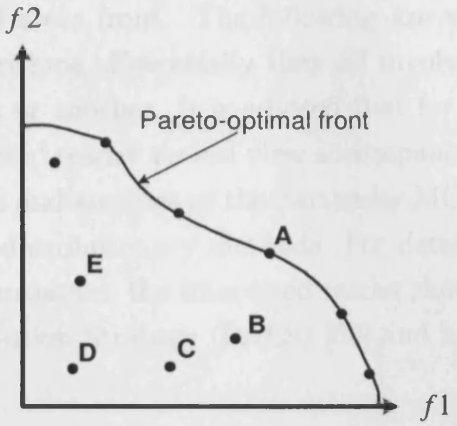



Figure 2.3: Pareto Optimal front

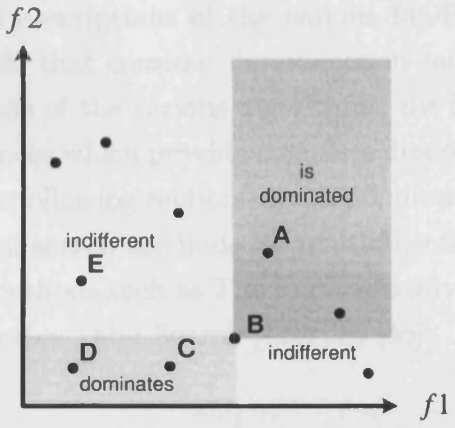



Figure 2.4: Pareto dominance regions with respect to B

### 2.3.2 Multi-Objective Techniques

As seen in section 2.2.7, multiple objectives can be combined to form a linear combination but the weighting process itself can produce ill-informed results. The GA is particularly suited to multiple objectives because the concept of pareto dominance can be incorporated into the GA's fitness operators [32]. How this pareto dominance fitness evaluation is performed varies according to MOGA technique but all methods attempt to spread the pareto optimal solutions along the pareto front. The GAs ability to search complex tradeoff surfaces also make them particularly suited to multiobjective techniques where these surfaces can become complex. Many multiobjective GA techniques such as [50], [32], [88], [26] incorporate their method primarily in the fitness selection procedures of the GA.

The Multi-Objective Optimisation Problem (MOP) uses the notion of a pareto dominance (section 2.3.1) to evaluate and drive the optimisation. The MOP can be expressed using these basic concepts:

1. Form a set of solutions on the Pareto optimal front in a single simulation run.
2. Maintain algorithm performance (selective pressure) and progress estimated Pareto front towards optimal front.
3. Produce a good distribution of solutions along the Pareto optimal front.

This leads to two questions which multi-objective Evolutionary algorithms (MOEA) attempt to solve:

1. How to assign fitness and then select appropriate solutions to drive the Pareto front?
2. How to produce an even representation of the Pareto front?

Selection and diversity of the solutions is therefore crucial in not only maintaining MOEA performance but also in selecting solutions that are representative of

the Pareto front. The following are very brief descriptions of the various MOEA algorithms. Essentially they all involve methods that consider dominance in some form or another. It is advised that for full details of the various algorithms, the interested reader should view accompanied references which provide complete descriptions and analysis of the particular MOEA. The following sections detail population based evolutionary methods. For details of local search methods for multiobjective optimisation, the interested reader should see methods such as The Pareto Archived Evolution Strategy (PAES) [60] and Multiobjective Tabu Search (MOTS) [46].

### VEGA

Schaffer's Vector Evaluated Genetic Algorithm [88] forms a set of representative solutions by repeatedly switching between the competing objectives. This process is done until there is an even proportion of solutions in the separate objective mating pools. The mating pools are then aggregated and selection/mating performed as usual. This algorithm has been improved by Fonseca and Flemming [33] to overcome problems with the algorithm but the original concept remains. An outline of the algorithm is as follows:

1. Set the initial population $P_t$ and begin with an empty mating pool.
2. For each individual $i \in P_t$ calculate fitness with respect to each objective.
3. Select the best individuals in each objective up to N/k where N is the population and k is the number of objectives.
4. Merge mating pools of both objectives, perform crossover and mutation.
5. Perform selection and repeat.

A graphical representation is shown in the figure below where the best individuals in each dimension are chosen for reproduction. Then mating pools are merged and shuffled, and crossover and mutation are performed as normal. Problems have been identified with the algorithm [33], [94] mainly because the the algorithm's tendency to prefer solutions towards the extremes of each objective leaving central regions of the pareto front unexplored.

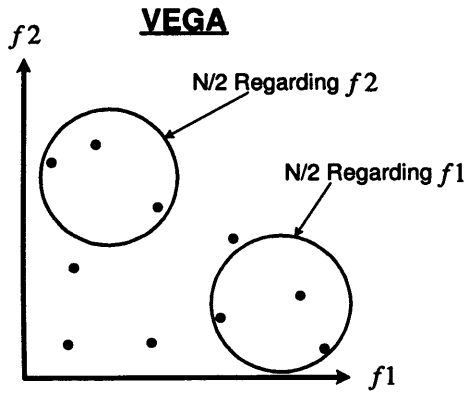



Figure 2.5: Vector Evaluated Genetic Algorithm

### NSGA

Niche Sorting Genetic Algorithm [94] uses the concept of evaluating solutions based on different tradeoff surfaces. These surfaces are then compared and selection is based on the coverage of each surface. A niche is also formed whereby each solution is compared to other solutions within some given distance. The niche is gradually made smaller to gain better granularity along the Pareto front. The fewer solutions within the niche, the more likely niche members will be selected. An outline of the algorithm is as follows:

1. Set $P_{Remain}$ to incorporate the entire population $P_t$.
2. Evaluate non-dominated individuals $P_{nondom}$ and give them dummy fitness (front value). Remove $P_{nondom}$ values from $P_{remain}$. After front removal, $P_{Remain} = P_{Remain}$ - $P_{nondom}$.
3. Perform Niching among fronts' members.
4. Perform selection according to niche and dummy fitness.
5. Perform crossover and mutation.

The algorithm can be best described using a flow diagram as shown in 2.7. The primary role of the algorithm is to form the relative surfaces and then form niches at each of the surfaces to allow even selection pressure along the pareto surfaces. A graphical representation of the front or surface formation is shown in figure 2.6 below. Full details of best practice for niche formation and full details of the algorithm can be found in [94].

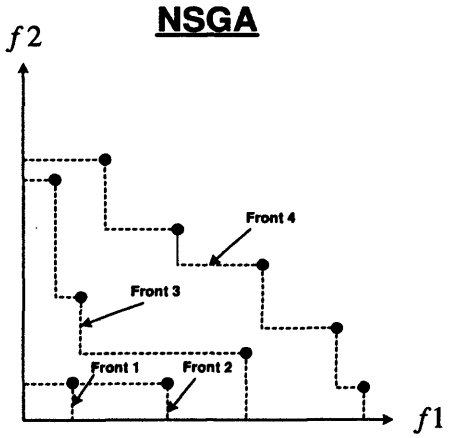



Figure 2.6: Pareto Optimal front

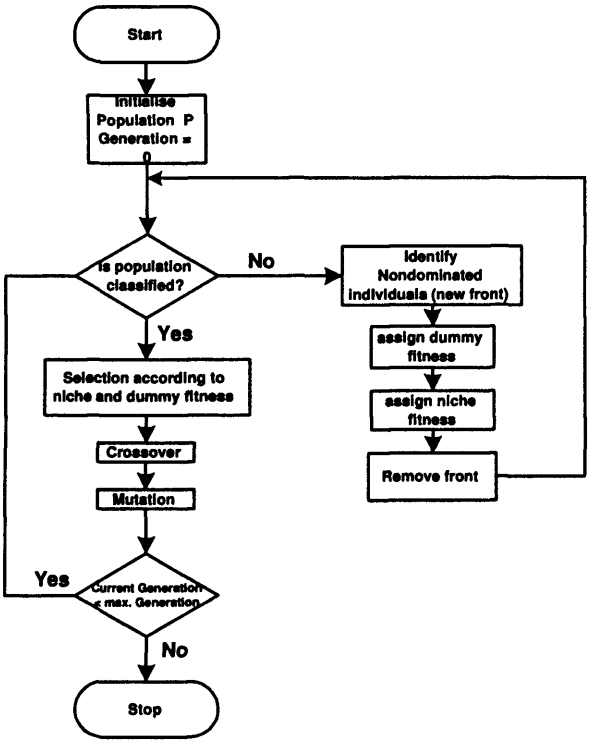



Figure 2.7: NSGA flow diagram

### FFGA

Fonseca and Flemings Multiobjective Genetic Algorithm [32] uses a fitness assignment that calculates the fitness based on the number of solutions that it dominates. An outline of the algorithm is as follows:

1. For each member $i$ of the population $P_t$ calculate its rank $r(i)$ based on the number of solutions that it is dominated.
2. Sort population according to rank.
3. Average the fitness values of individuals with the same rank to ensure that all of them are sampled at the same rate.

A graphical representation is shown in figure 2.8 below. The corresponding ranks are shown in the figure. The original method shown here has been adapted using more complex niching mechanisms [34] to improve exploration of the pareto front.

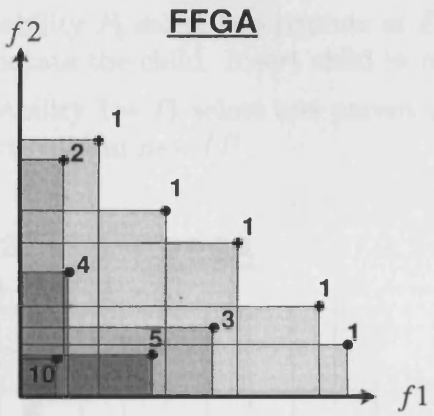



Figure 2.8: Fonseca and Flemings Multiobjective Genetic Algorithm

### SPEA

Strength Pareto Evolutionary Algorithm [26] is a mixture of established and new concepts. It uses a ranking type procedure based on dominance and an "External Set" structure to control Pareto front evaluation. It is in the newest class of multiobjective genetic algorithms that utilise elitism to improve performance. The operation of the algorithm is detailed in section 4.3.

### PESA

Pareto Envelope-based Selection Algorithm [17] uses the concept of a hypergrid to control diversity and selection. PESA is unusual in that, although it uses dominance to form a pareto front, it does not use the degree to which one solution dominates another. It uses the hypergrid alone to progress the pareto front. Another difference is that it only selects individuals from the "external" set, shown in figure 2.9 by the circular non-dominated members (dominated members are shown as black squares). An outline of the algorithm is as follows:

1. Generate a population $P_t$ and initially insert all members in the internal population $IP$. Initialise external set $EP$ to an empty set.
2. Move all non-dominated members of $IP$ to $EP$
3. Delete all current members in $IP$. Then, until maximum number of generations or some termination criteria is reached, repeat:
   - with some probability $P_b$ select two parents in $EP$, produce a child with crossover and mutate the child. Insert child in new $IP$.
   - with some probability $1 - P_b$ select one parent in $EP$, mutate and form new child. Insert child in new $IP$.

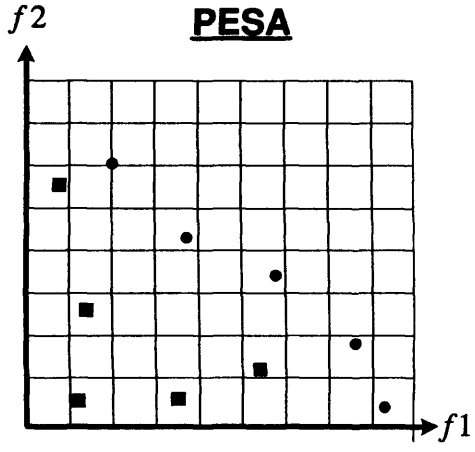



Figure 2.9: Pareto Envelope-based Selection Algorithm

Selection pressure is maintained by scoring members using what is termed the "squeeze" factor. The external set members are scored based on the number in the same grid, the more external set members in a grid, the lower the score. This allows selection pressure to be maintained in areas that have fewer external set members.

All the previous algorithms are limited in dimensionality. Most are described based on the assumption that there are only two competing criteria. A number of these techniques have been adapted to allow multi-dimensional evaluation but it has been suggested that as the dimensionality increases, performance decreases [33]. This thesis uses only two criteria simultaneously.

It is also important to note that despite the variety of different techniques, evaluating their relative performance is still very difficult [49]. Horn et al[49] does suggest that, in general, there are no superior algorithms for all classes of multi-objective optimisation. This therefore implies that it is difficult to say categorically which algorithm is better for any application, other than through empirical tests for any single application.

### Why SPEA?

As stated in section 1, the primary aim is to determine if radar network design can be achieved using an optimisation method. So, whilst it is important to choose a good optimisation method, the performance is not the primary concern, providing it gives some indication of the radar network design possibilities that can be achieved.

It is clear that the elitist methods such as SPEA and PESA have been proved to outperform more traditional MOGAs [26] such as VEGA, NSGA and FFGA. Therefore it would seem prudent to use an elitist method, either PESA or SPEA.

Although the principles involved in both SPEA and PESA are very similar and in fact, the performance is very similar [17], SPEA is easier to implement and control. SPEA has an implicit fitness assignment which requires no parameter setting. PESA, however, uses a hypergrid structure which, while simple, requires more complex parameter setting in order to get the hypergrid squares the correct size. Although the author of [17] does suggest an automatic hypergrid sizing and resizing method, it is unclear as to how well this performs on all classes of problems.

For simplicity of implementation, parameter setting and performance, SPEA was chosen as the MOGA to use in conjunction with the radar network design problem.

CHAPTER

# THREE

# Modelling and Simulation

This section outlines the role simulation and modelling have as a core part of this thesis. It also outlines the design and implementation considerations of the simulation framework.

Simulation represents the evaluation stage of this thesis, see figure 1.3 page 28. Simulation more traditionally provides a method to analyse system performance under operating conditions. The nature of this thesis shifts the emphasis from merely simulating the operation of the network to more fundamental aspects of the radar communication system itself. The unrefined nature of the radar communication model requires a flexible simulation model to allow refinements to be made.

When compared with wireless networks, the radar networks involved in this thesis can be considered infrastructure-less ad-hoc networks [76] as they act as both radar nodes and network infrastructure.

The purpose of the simulator in this thesis is to allow the radar network designs to be critically analysed. This requires firstly simulating the target processing and the subsequent message production to allow accurate traffic modelling. Then a radar communication protocol is simulated to model the packet/message flow from the sending radar node to the commander radar. The entire process will therefore allow the simulation of the transfer of packets around the established radar network in response to a target or series of targets.

## 3.1 The Network Simulator

Performance modelling and simulation have become a central issue in computer science. [102] provides a detailed account of the theory, engineering and applications of network simulators.

There is a need to provide realistic, quantative analysis of the designed radar networks in order to evaluate their performance. In addition, the simulator needs to be flexible in order to reflect changes in hardware/operational aspects of the problem.

The radar network simulator is required to simulate the the interaction between the radar and targets. Its is also required to simulate the interaction of radar-to-radar communications in a radar network. All messages generated by the radar nodes in response to targets must be sent to a commander (positioned at a single radar node anywhere in the radar network) via the radar network. This message passing/packet routing must be simulated in the form of a routed network to allow the full evaluation of the radar network designs. In order to perform this function the simulator should be able to:

- ***Represent the physical aspects of the environment, lines-of-sight, terrain, radio propagation model.*** This aspect of the simulator is incorporated in the Geographic Information System detailed in section 3.4.

- ***Represent the physical aspects of radars, radar node operations and links.*** Ranges, communications timings, and data rate characteristics of the radars need to be simulated.

- ***Allow adaptations to the physical aspects of the radar communication nodes.*** The physical characteristics should be adaptable to allow different methods of operation to be evaluated. This allows modifications such at number of links per face (section 3.2.4).

- ***Provide abstract representation of the radar.*** The radar operation is simulated through the generation of messages based on target proximity rather than true radar operation based on the electromagnetic properties of the system.

- ***Use suitable network level protocols including routing.*** The routing of messages is carried out using routing tables maintained by each radar node. The routing is detailed in section 3.2.4.

- ***Allow quantative and qualitative information to be extracted from the various components of the system.*** Each message is tracked to allow message delays, queueing, and overall message times to be analysed.

### 3.1.1 Current Network Simulators

There are various network simulators available. The most popular of these are listed in [31]. It lists the most widely used commercial and academic network simulators, the most widely used of these being NIST [1] ns-2 [2] and OPNET [3] The network simulators in Table 3.1 offer substantial support for modelling traditional fixed networks and their protocols. They have more recently added support for wireless networks. A key aspect of this project is being able to represent the nodes in spatial terms. This is critical to this project as both the network topology and message distribution are governed by node position. In the above table only OPNET currently uses the notion of node position. All the network simulators in Table 3.1 provide good support for IP oriented and packet based protocols.

| Simulator | REAL | ns-2 | OPNET |
|---|---|---|---|
| Operating system | UNIX/LINUX | UNIX/LINUX | Windows |
| Wireless network support | No Support | Limited Support | Supported |
| Protocols | IP Orientated | IP Orientated | IP Orientated |
| Discrete event simulation | Yes | Yes | Yes |
| Packet-switched communication | Yes | Yes | Yes |
| Physical aspects of wireless links | No Support | Limited Support | Yes |
| Radar representation | No Support | No Support | No Support |
| Reliability metrics (e.g. connectivity) | No Support | No Support | Limited Support |

Table 3.1: Comparison of selected network simulators

Our problem could be simulated using OPNET as it provides good support for wireless networks. However to date, none of the simulators available, including OPNET, combine network simulation with the operational characteristics of a radar. Lofquist [63] describes the use of radars in inter-continental radar systems but uses a predefined traffic matrix (modelled using OPNET) rather than an event driven model. Incorporating operational complexity of the radar within the simulator means that only a custom built network simulator will be able to provide adequate support for this project. To that end a custom built network simulator and evaluator is used.

[1] Available from NIST www.snad.ncsl.nist.gov/itg/nistnet/
[2] Available from VINT http://www.isi.edu/nsnam/vint/
[3] Available from OPNET http://www.mil3.com/

## 3.2 The Radar Node

The radar node simulates the basic operations of a radar based on the proximity to a target. Figure 3.4 is representative of how targets and radars interact. The nodes also contribute to the formation of an infrastructure-less network and can be considered ad-hoc [76]. The nodes therefore perform the following ad-hoc and radar functions:

- Operate as a radar and generate information and messages regarding targets as shown in figure 3.4.
- Operate as a router and forward packets using the established radar network.
- Establish directional beams to facilitate communication and formation of an radar network.

### 3.2.1 Radar Node Characteristics

Table 3.2 shows a summary of the important characteristics of both the radar and communication characteristics of each node. Radar specific metrics such as data-rate at separation, have been obtained through empirical testing by Qinetiq Malvern.

| Attribute/Characteristic | Value/ Description |
|---|---|
| Power (mW) : | 200 |
| Frequency (GHz) : | 15 |
| Wavelength (m) : | 0.02m |
| Radar Communication Range : | 7500m |
| Radar Target Range: *Surveillance mode* | 3750m |
| Radar Target Range: *Tracking mode* | 5000m |
| Radar ID | 12Bit ID |
| x-Coord | OS Reference |
| y-Coord | OS Reference |
| z-Coord | Elevation |
| Data-rate at 1km Separation | 2MBit/s |
| Number of Phased Arrays | 4 |
| Max No. Acquired Targets | 20 |

Table 3.2: Radar and Communication Characteristics of a Node

### 3.2.2 Radar Node Subsystem

Figure 3.1 shows the message processing and associated simulator subsystems of the radar node.

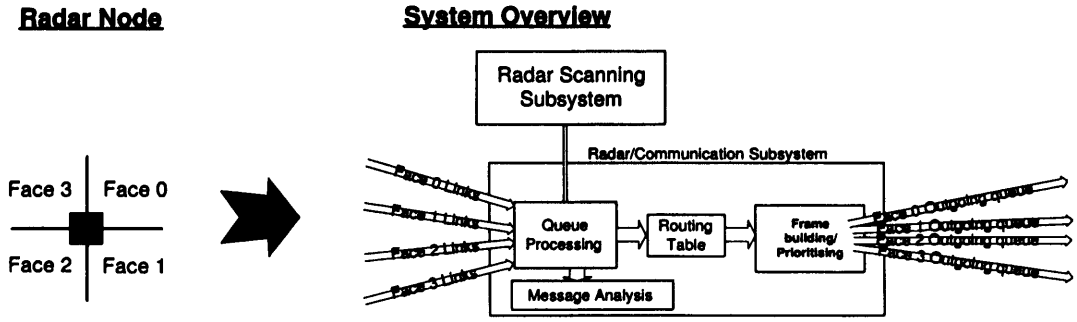



Figure 3.1: Radar node overview

Messages can arrive either from other radar nodes through the incoming links or directly from the radar subsystem. Incoming messages are then processed depending on whether incoming messages are destined for that node or another node on the network. Messages destined for that node are analysed by a separate subsystem. Messages/packets that are destined for a different node on the network are rebuilt/reframed using the relevant IP protocols or radar network specific protocols. They are then put on an outgoing message queue dictated by the routing table. If required, the outgoing queues can be prioritised depending on message type. E.g. critical or more time sensitive messages messages could be placed at the front of the queue and surveillance messages moved to the back of the queue.

Queueing is based on a First-Come-First-Serve method unless otherwise stated. Other queueing methods such as those discussed in Tanenbaum [96] and more detailed algorithms such as rate-controlled algorithms have not been considered.

### 3.2.3 Radar Node Message Design

Messages that are sent to the commander contain information regarding acquired targets. Rather than simulate the data-link and network protocol framing[95], representative frame bits have been added instead. The network headers are based on a reduced form of TCP/IP headers (compressed TCP/IP headers) [54]. Data-link headers are based on the Point-to-Point Protocol [92]. Compressed headers are being used to conserve bandwidth and avoid complexity not required with full TCP/IP headers.

Although the message passing in this project is not connection-orientated, future simulation might require the options at the transport layer allowed using the compressed TCP/IP headers. The interactive nature of target engagement would require connection-orientated options to accurately engage/track the target.

| Frame Description | Frame Fields | No. of Bits |
|---|---|---|
| **PPP Data Link Frame Header** | Flag | 4 |
| | Control | 8 |
| **Compressed IP Header** | Bit Changed Flags | 8 |
| | Sequence No. | 4 |
| | Control Flags | 8 |
| **Radar Frame** | Radar ID | 12 |
| | Destination Radar ID | 12 |
| *Target information* | Time | 16 |
| | Range | 18 |
| | Az | 12 |
| | El | 12 |
| | Velocity | 10 |
| | D Range | 8 |
| | D Az | 8 |
| | D El | 8 |
| | D vel | 8 |
| **PPP Data Link Frame Trailer** | CRC 16 | 16 |
| | Flag | 8 |
| | Total Number of Bits | 180 |

Table 3.3: Message specification and associated headers

### 3.2.4 Radar Communication and Operation

The radar node itself is made up of four phased arrays which each cover 90 degrees. Note that the radar range to the target is much less than the radar communication range (not shown in figure 3.2). See section 1 for full details. The radar nodes are able to communicate if they have available timeslots and are separated by less than 7500m as stated in section 3.2.3.

Figure 3.2 shows how the communication model is intertwined with the radar model. In figure 3.2 when node 1 detects a target it sends messages to or via node 2 and onto its final destination. The radar performs normal radar function for 90% of the radar time and enters a communication mode for the remaining 10% of the time. Each 100ms timeslot is split as shown in figure 3.2, into 90ms radar scan followed by 10ms of communication. Node 2 face 3 has to be pointing in the correct direction and at precisely the the same time during communication between node 1 face 1.

This highlights the precise time synchronous nature of the operation of the radar nodes. Not only does the beam from node 1, face 1 need to be pointing at exactly the right angle to node 2 face 3 but they have to be pointing at each other at precisely the same time. Otherwise the receiving beam would be expecting to receive information when none was being transmitted and vice versa.

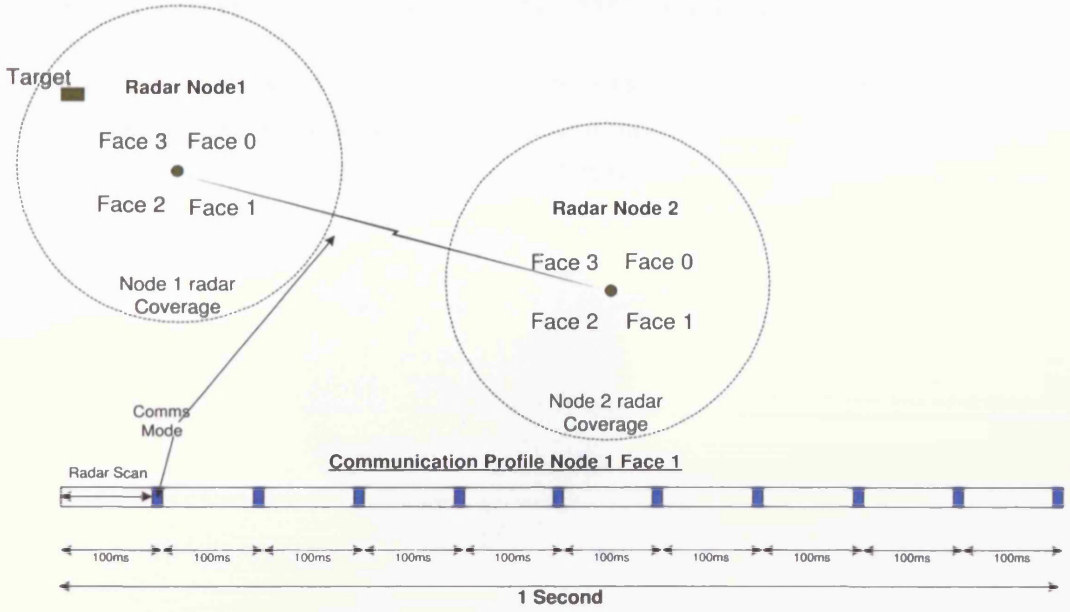



Figure 3.2: Radar Operation 1 link per face

The radars can have up to 2 communication links per face (also abbreviated LPF in this thesis). This requires the timeslots to alternate between the different nodes. If, as in figure 3.3, node 1 sends messages to both node 2 and 3 on the same face, then the communication time is split as shown.

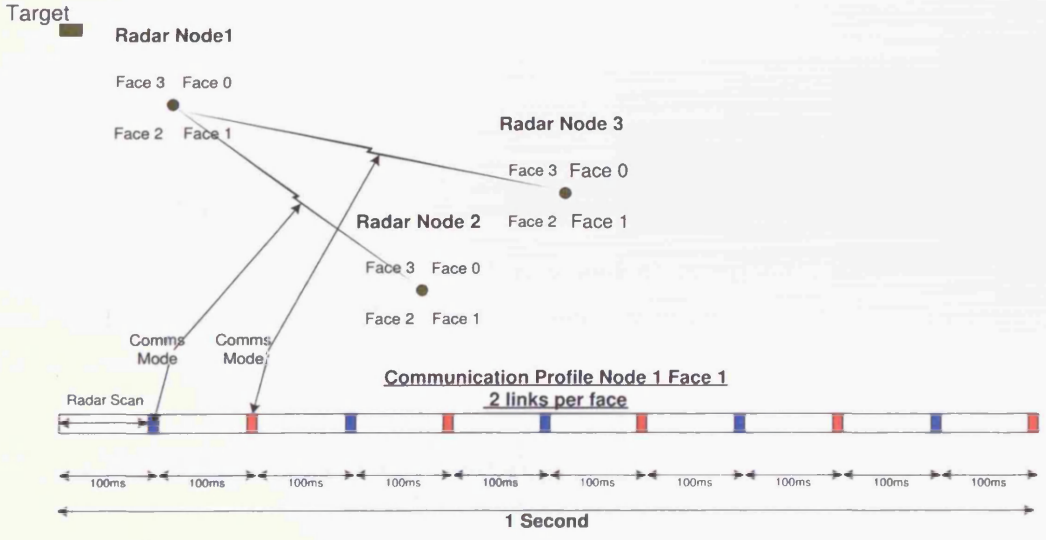



Figure 3.3: Radar Operation 2 links per face

## 3.3 Discrete Event Driven Simulator

Discrete events imply by definition that the simulator is run based on a series of specific events. The simulator generates communication messages based on the proximity of a target. An example is given in figure 3.4.

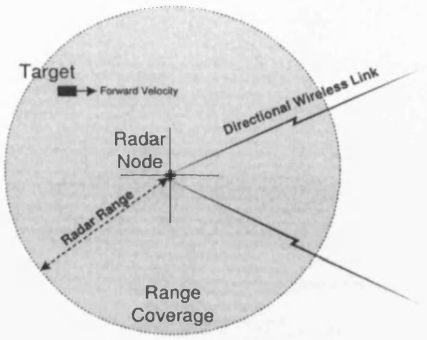



Figure 3.4: Discrete Event Generation

At time $t$ an unknown target travelling at velocity $v$ is detected by radar $r$. This *event* triggers the radar node $r$ to carry out operations required to monitor and identify the target. Surveillance and tracking messages are then built and sent to the required node via an already established wireless link (wireless link is only available during designated timeslots, see section 3.2.4). The node receiving the messages is the commander of the system located at some position within the network.

As targets travel through the network, each node attached to the network detects targets within their range and sends the relevant messages to the commander node.

### 3.3.1 The Target

Targets in this thesis are given by a set x, y and z[4] component of velocity. The velocity of the projectile is set initially and remains unchanged during the simulation. Table 3.4 outlines the properties of the targets.

| Attribute/Operational State | Description |
|---|---|
| TargetID | For simulation use only |
| Velocity | x, y, z components of velocity |
| Current Position | Current position calculated based on simulator time |
| Entry Point | Starting position of the target |

Table 3.4: Target Attribute/operational details

[4]Z-component of velocity included for completeness

Any number of targets can enter a designated airspace. Throughout this project the straight line speed of all targets is set at 250 m/s for comparison purposes [83]. The 28km x 28km grid would have a maximum of 14 targets [83].

## 3.4 The Environment

The radars are placed in a simulated environment using a Geographic Information System. A Geographic Information System (GIS), is a computer-based modeling tool for mapping and analyzing things that exist and events that happen on earth and its environment. The GIS integrates common database operations, mathematical modeling techniques and visualisation techniques to accurately model, analyse and visualise an earth system. In this thesis spatial characteristics such as distance, elevation, terrain and lines-of-sight can be accurately modelled using the GIS.

As the nodes in this project are highly directional, accurate lines-of-sight are vital. Lines of sight are calculated using the GIS system ArcGis[5]. It allows for the customisation of lines of sight calculations. The terrain datasets used are digital elevations models taken directly from the Ordnance Survey (OS). The datasets in NTF[6] format high resolution terrain models and provide accurate models of the terrain. OS grid SH (where SH is the alphabetical grid reference system used by the Ordnance Survey) is used throughout this project unless otherwise stated. The grid is shown below in Figure 3.5 along with a radar coverage map of 25 radars as an illustration.


Figure 3.5: Terrain map with coverage map of 25 radars

[5]See http://www.esri.com/software/arcgis/overview.html for full details of ArcGIS
[6]Ordnance Survey PANORAMA Digital Terrain Model

The radar nodes are placed on the grid either randomly or at predefined positions and the GIS calculates the lines of sight based on three criteria:

1. Radar range
2. Node height
3. Interference/ propagation [7]

This Preprocessing stage of the system is shown in Figure 3.6

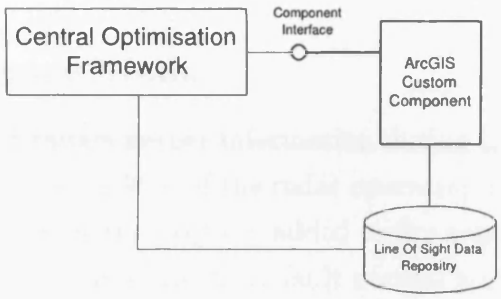



Figure 3.6: Preprocessing

The central framework encapsulates the entire system. The system is given access to the GIS Component[8] at all times during both the optimisation and simulation stage. The framework can initially call for current lines-of-sight for a given set of nodes or retrieve previously calculated lines-of-sight. Calculations of lines-of-sight on-the-fly require enormous computational power. Therefore the calculations are made based on predicted node movements and therefore lines-of-sights can be precomputed. Table 3.5 below gives an indication of CPU times (see section 3.8 for simulation platform details) for different numbers of nodes.

| Number of Nodes | 16 | 32 | 64 | 128 | 256 |
|---|---|---|---|---|---|
| CPU Time (seconds) | 12.4 | 47.4 | 142.72 | 694.19 | 2484.15 |

Table 3.5: Lines-of-Sight Calculation

The table above shows how difficult calculating the lines of sight for all nodes becomes as the number of nodes increases. This is because each node pair line of sight has to be calculated which is an $O(N!)$ operation.

[7] ArcGIS solution developed includes interfaces for the inclusion of a propagation model

[8] The GIS component conforms to the Microsoft Specification for binary COM objects http://www.microsoft.com/com/

## 3.5 Traffic Generation

Section 3.3 describes the triggering event of a target passing within range of the radar. This section[9] describes the three modes of operations and the timings involved with:

- Surveillance Mode
- Tracking Mode
- Engagement Mode

### 3.5.1 Surveillance Mode

It is assumed that the radars gather information during the scan phase of the radar operation which accounts for 90% of the radar operating time. After a target is first detected within radar range the target is added to the acquired target list. The node changes to surveillance mode from the default normal state. The range is increased to reflect this change in mode to 1.5 $\times$ minimal radar range.

After a confidence period of 2 seconds has elapsed, the target is tracked to ensure that it is not simply noise. Messages regarding the target are sent to the commander via the network at 1 second intervals. After a further 5 detections/seconds the radar node changes to tracking mode.

[9]Developed in accordance with Qinetiq recommendations

### 3.5.2 Tracking Mode

In tracking mode the range is further increased to 2 × minimal radar range to focus the node effort on tracking the acquired targets. The number of messages sent to the commander is therefore increased to 2 messages per second (1 per 0.5 seconds). Figure 3.7 shows how a target enters the radar range of node 16. After 2 detections it will increase its range and enter a surveillance mode and send messages as required. After a further 5 detections, it increases its range further and enters a tracking mode.

The commander node assumed to be at node 23, on receiving tracking messages, sends messages to nodes along the predicted path. In figure 3.7 these would be nodes 14, 15 and 13. These nodes then all increase their radar target range to a tracking range of 5000m (see radar characteristics section 3.2) in readiness for the predicted approaching targets. These radars now know the approximate velocity and arrival time at their node and extend their range in anticipation of the approaching taget[10].

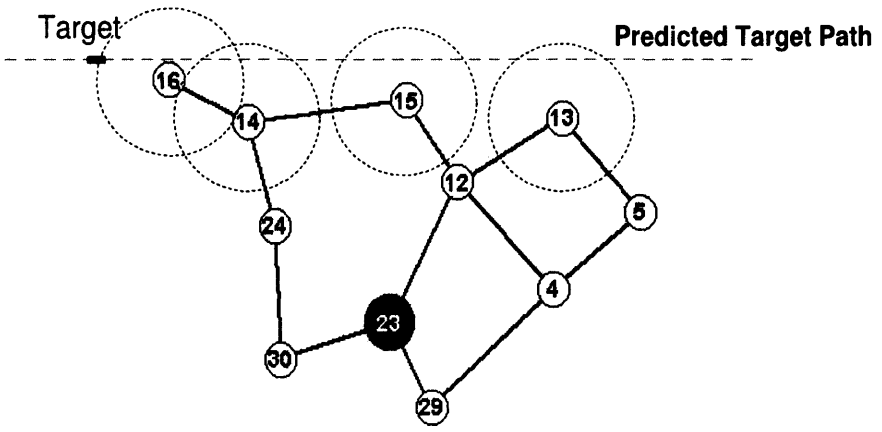



Figure 3.7: Tracking operation (radar target ranges shown)

### 3.5.3 Engagement Mode

After the target has been tracked for some period of time, dictated by the time a commander takes to make a decision to counter the target, the commander issues an order to engage the target ie. fire a missile to intercept the incoming target. This involves updating the missile with new target information during the interception

[10]In reality the tracking mode would not increase range but increase dwell time for that particular area in space. The simulator extends range simply to increase the number of target messages produced as a real tracking mode would

and increased detections of the target. This is a multifaceted problem and is not considered in this thesis.

## 3.6 Graph Colouring

Graph colouring is an important aspect of the simulator. The communication between nodes relies on each link being setup in the correct timeslot. The correct timeslots need to be synchronised. This requires the use of edge-colouring the graphs to ensure that the nodes are properly synchronised. Figure 3.8 shows a simple three node network. Node 1 on the left network has two red timeslots allocated and requires that it communicates between both nodes 2 and 3 in the same 10ms allocated timeslots. Obviously this is not possible as only one directional beam can be formed at any one time. Node 1 on the right has both a red and a blue timeslots allocated. During the 10ms blue timeslot its communicates with node 2 and during the red 10ms timeslot it communicates with node 3. By edge-colouring the graph and faces, this clash of timeslots is avoided by ensuring that only possible timeslots are allocated to communicating faces.

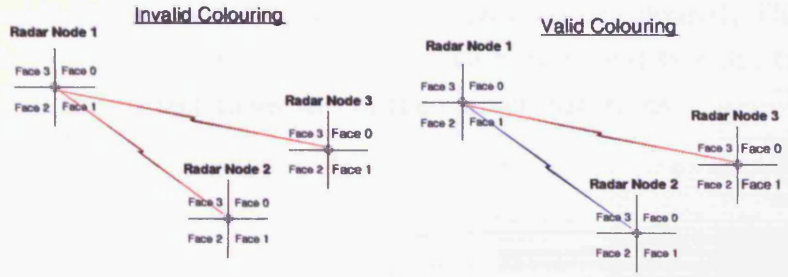



Figure 3.8: Tracking operation

Each timeslot and associated colour represents a 10ms connection between 2 nodes as indicated in figure 3.3. The number of timeslot colours is dictated by the number of links allowed per face. If each radar face is allowed two links then two distinct timeslots are required. If the allowed number of links per face is two, then the number of colours in the palette of the colouring algorithm is two.

As the graph colouring problem is a classically computationally hard mathematical problem [91], it was decided to use a fast randomised algorithm. The inherently tight upper and lower bounds make finding an optimal colouring less important [100]. Because time is the critical aspect of this project, a highly randomised algorithm using the Nibble method is used [25].

### 3.6.1 The Nibble Method

The nibble method is a randomised graph colouring algorithm. It can be used as either a distributed algorithm or a global algorithm where all edge information is known.
The algorithm proceeds as follows [11]:

1. ***(Select nibble)*** Each vertex $u$ randomly selects a fraction of the edges $e$[12] incident on itself, 1 edge in this project. An edge is considered selected if both of its vertices select it.
2. ***(Choose tentative colour)*** Each selected edge $e$ chooses independently at random a tentative colour $t$ from its palette of currently available colours. The available colours are the maximum number of link colours. In this particular radar application it is limited to two colours. However any number of colours can be used.
3. ***(Check colour conflicts)*** Colour $t$ on each edge $e$ becomes the final colour of $e$ unless some edge incident on $e$ has chosen the same tentative colour.
4. ***(Update graph and palettes)*** The graph and the palettes are updated by setting the palette of vertex $u$ minus the already allocated colours.

This procedure is then repeated until no more edges can be coloured. This colouring is then applied to the network, ensuring that each node and face are transmitting and receiving at the correct times and in the correct directions.

[11] Full details available in section 4 [25]
[12] This fraction determines the speed of the algorithm. Lower numbers have better performance

### 3.6.2 An Illustration

Figure 3.9 below is a simple example of how the nibble algorithm edge-colours a graph.

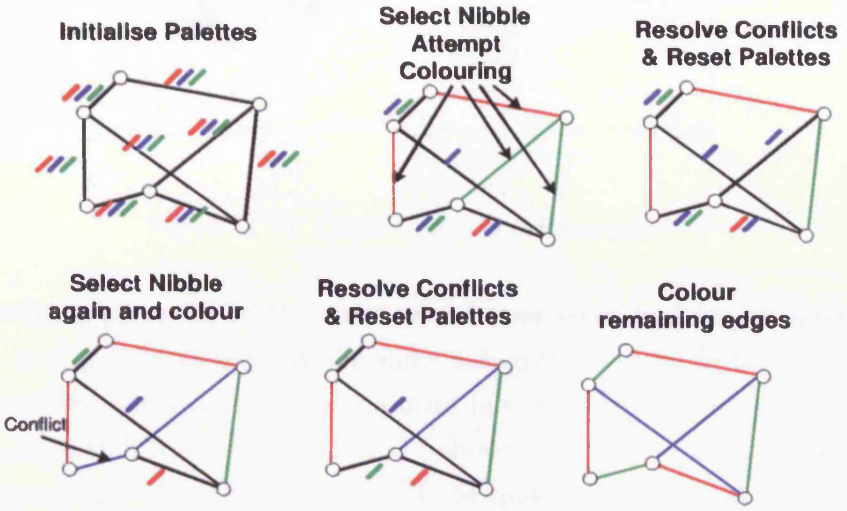



Figure 3.9: Nibble Step-by-Step Example

### 3.6.3 Quadrant Colouring

The networks considered in this thesis do not require all adjacent links to be different colours as in the traditional edge-colouring algorithms. Only links on the same face require proper edge-colouring. Therefore a series of subgraphs containing all adjacent edges on the same face require edge colouring.

Take as an example figure 3.9 page 70, say node 0 of graph G (shown on the left) needs to synchronise face 1 (shown on the right) with all adjacent nodes in face 1. This is then carried out for each node link on face 3 of node 2. In this case neither node 2 nor node 3 has any other corresponding links in face 3 so the process ends. This results in a sub-graph of G (shown on the right). This three node subgraph is then coloured using the nibble method.

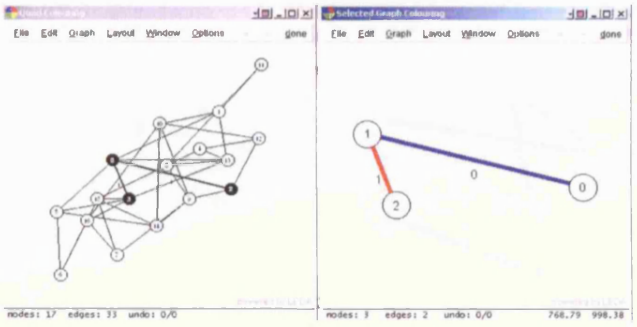


Figure 3.10: Quad Colouring Example

The algorithm proceeds in the same way for all faces and for all nodes. Obviously this is a greedy method as the same sub-graph will be formed by starting the algorithm from the same face, or opposite face, in the sub-graph. For example starting the algorithm from node 12 face 3 ( shown on the left of figure 3.11) would form the same graph as node 10 face 1. If the quadrant is already coloured then the resulting subgraph is compared to the previous colouring to determine the best colouring.

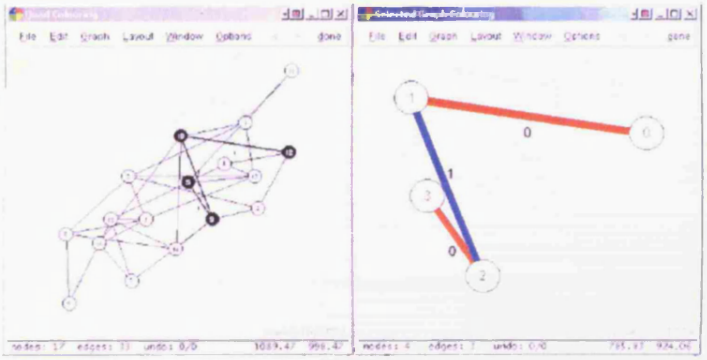


Figure 3.11: Quad colouring example 2

Take another example shown in figure 3.12, consider node 10 (shown on the left of figure 3.12), by synchronising node 10 face 1, it causes the algorithm to proceed to form an odd cycle sub-graph. Obviously an odd cyclge is not 2 colourable. Either the link has to be removed or reallocated when using traditional edge colouring [100].

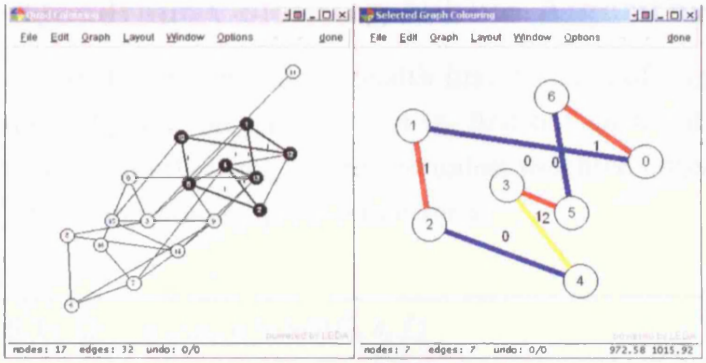
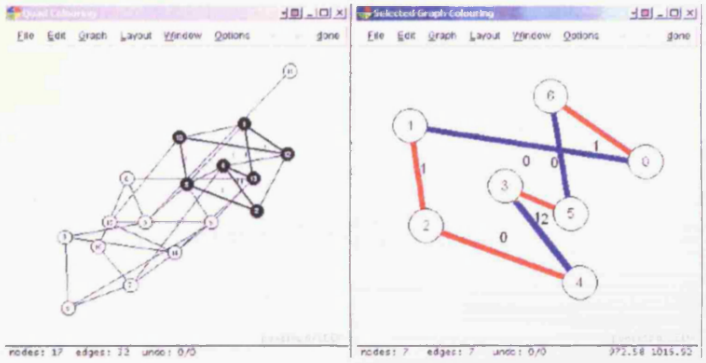



Figure 3.12: Quad Colouring Example 3

However, in our case, only the faces have to have separate colours so both links between node 2 and 1 and both links between 2 and 4 can have the same colour as shown in figure 3.13(right most graph) as they are on separate faces.

Figure 3.13: Quad Colouring Example 4



### 3.6.4 The Quadrant Colouring Nibble Algorithm

The algorithm is essentially a modified breadth first[13] search of a graph G starting from node $s$, face $f$. Q is a conventional first-in, first-out queue, $d$ is the distance from $s$ (distance not currently being considered unless weighted edges are considered at a later date) $\pi$ is a maintained predecessor for $s$.

---

**Algorithm 3.6.1:** QUADCOLOURING$(G, s, f)$

---

Radar Node Set V(G) /* Set of V radar nodes in Radar network G */
node u /* Current selected node */
colour[u] /* The color of each vertex $u \in V$is stored in $colour[u]$*/
$\pi[u]$ /* The predecessor of $u$ is stored in variable $\pi[u]$*/
$d[u]$ /* The distance from source $s$ to vertex $u$ is stored in variable $d[u]$ */

**for each** $u \in V[G] - s$
**do** $colour[u] \leftarrow WHITE$
$d[u] \leftarrow \infty$
$\pi[u] \leftarrow NIL$
***end for***
$d[u] \leftarrow 0$
$Q \leftarrow [s]$
**while** $Q \neq empty$
**do**
$u \leftarrow head[Q]$
**for each** $v \in adj[u]$ **and** $v \in adj[f]$
**do**
**if** $colour[v] \in WHITE$
**then**
$colour[v] \leftarrow GREY$
$d[v] \leftarrow d[u] + 1$
$\pi[u] \leftarrow v$
$ENQUEUE(Q, v, (f+2)mod4)$
$DEQUEUE - Head(Q)$
$colour[u] \leftarrow BLACK$
**RUN NIBBLE on SUBGRAPH**

***Functions:***
$ENQUEUE(Q, v, (f+2)mod4) \leftarrow$ Add node v to current queue Q and mark correct face
$DEQUEUE - Head(Q) \leftarrow$ Removes the node v at the head of queue Q
$head[Q] \leftarrow$ Returns nove v and the head of queue Q
$adj[u] \leftarrow$ Returns all nodes v that have links connecting u
$adj[f] \leftarrow$ Returns all nodes v that have links connected on face f

---

[13]More detailed analysis of depth first type algorithms can be found in [16]

## 3.7 Routing Algorithms

The main function of a routing algorithm is to get a message/packet from a source node to the required destination node [96]. It is also responsible for how it gets from a source node to the required destination node. This decision can be based on a variety of factors from very simple metrics such as hop count (defined formerly as Routing Information Protocol, RIP [47]) to more complicated routing protocols based on zones and node powers such as in Zone Routing Protocol,ZRP [44].

In order to simulate and evaluate our networks we require a network routing algorithm that is simple, efficient and is representative of the network topology. The simplest and most immediately obvious method would be to use RIP which is basically the minimum number of hops. The problem with RIP is that although it can reflect changes in the network topology, it cannot reflect rapid changes in network traffic without substantial overheads. Other protocols such as Open Shortest Path First OSPF [7] can incorporate many other metrics to decide the shortest path from source to destination based not only on hop-count but on other metrics such as distance, delay or queue-length.

The following sections illustrate some simple routing methods that improve the overall performance of the network by utilising link and routes more effectively. The routing algorithms used are to approximate how the messages might be routed to their destination in the real network. The following section does not detail the more sophisticated algorithms that might actually be implemented for this type of network. A detailed discussion of the performance comparison of some current state-of-the-art routing algorithms that would be used for dynamic networks can be found in [8].

### 3.7.1 Shortest Path Routing

Before discussing routing any further, shortest paths are defined. For a weighted graph $G = (V, E)$ where each edge $e \in E$ has an associated weight mapping $w(e)$, the weight of a path between a pair of nodes can then be defined The weight of a path $p = (u_0, u_1, u_2, ....., u_k)$ is the sum of its associated edges:

$$w(p) = \sum_{i=1}^{k} w(v_{i-1}, v_i)$$

The minimum path is then the minimum weight path w(p). Weights can then be associated with different metrics based on the application. If in the example shown in figure 3.14 each edge is assigned the weight of 1 then the shortest paths become minimum hop paths. The weights can also reflect other measures such as delay and queue length.

Routing can have a significant effect on the performance of a network design. Routing becomes more important as the size of the network increases. Even small networks such as those shown below in figure 3.14 can show improved performance when using efficient routing algorithms as packets can be routed along multiple paths to the same destination. There are however situations that cannot be entirely resolved by routing. In example 3.14 a simple method using shortest path routes are used to illustrate some problems that routing cannot overcome.

Figure 3.14 shows some oversubscribed links. The left value on each edge indicates the number of messages that can be sent in a single timeslot. The right values on each edge are the number of common shortest paths. The link between 14 and 29 can send a maximum of 4 messages in a single timeslot but the link has common shortest paths from 5 other nodes hence (4 / 5). Messages from 5,28,14,15, 36 are all sent along that link to the commander node. In this case the commander node is 0.

As node 0 only has 1 link, all other nodes will have to send all messages via that link. The link 25-0 is able to transmit 3 messages in a single timeslot. However it is along the shortest path from all other nodes (3/31). If, at any point, several messages need to be sent to the commander, this link will become congested. This example shows that although the routing might be able to route messages around the network efficiently, no routing algorithm will be able to compensate for lack of capacity at key points, namely the commander node. In this example, if the commander were placed at a node with higher capacity or multiple links, then there would be more routing opportunities and possibly higher capacity.

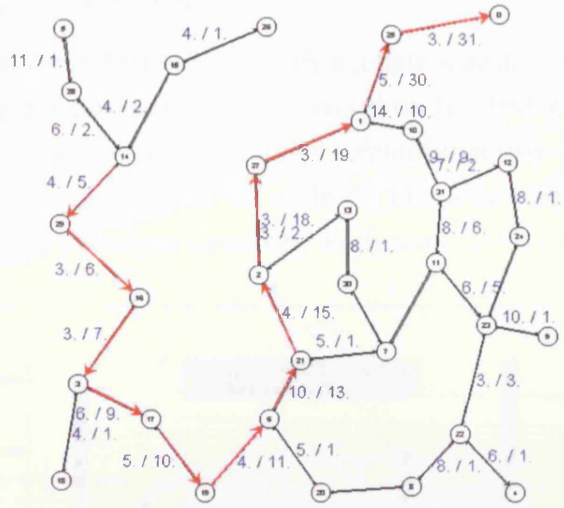



Figure 3.14: Graph of number of messages per timeslot and common shortest paths

### 3.7.2 Queue Based Routing

The queue based routing algorithm is identical to the shortest paths algorithm except that the metric that now defines the shortest path is queue length. The shortest path will be the path with least queueing at all intermediate node queues along the path. As in the previous section, the edge weights are set to reflect the current queue states. The shortest queue paths are then defined:

$$w(p) = \sum_{i=1}^{k} w(v_{i-1}, v_i)$$

except the weight $w(v_{i-1}, v_i)$ is equal to the queue length between $v_{i-1}$ and $v_i$. Weights are updated every 0.25 seconds to reflects changes in queue lengths. It therefore checks the queue lengths every 2 timeslots. The routing is based on a very simple procedure as set out below:

```
Initialise shortest paths
    Repeat every 0.25s
        build Distance Vector Routing Tables
        attach Queue Length values to distance tables
        compute new routes
        if better routes available
            update Routing Table
    Until end of simulation
```

It should be noted that these routing methods are used for simulation purposes only. Fully implemented algorithms have efficient route updating and maintenance procedures which reduce routing overhead [8].

### 3.7.3 Routing Test Case 1

The following graphs detail the results from a single simulation run using shortest paths which was then repeated using queue based on the test graph shown in figure 3.14 page 75. Tests were run with 14 targets simultaneously travelling across the network and the commander located at node 1. The following two graphs (figures 3.15 and 3.16) show the messages routed by each node to the commander.

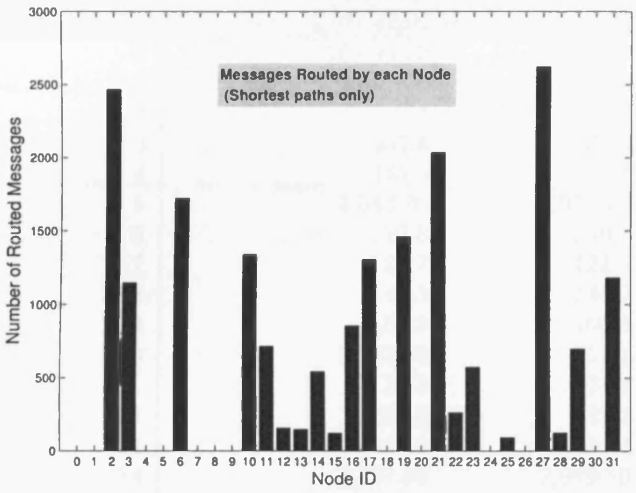



Figure 3.15: Message Distribution with shortest path Routing

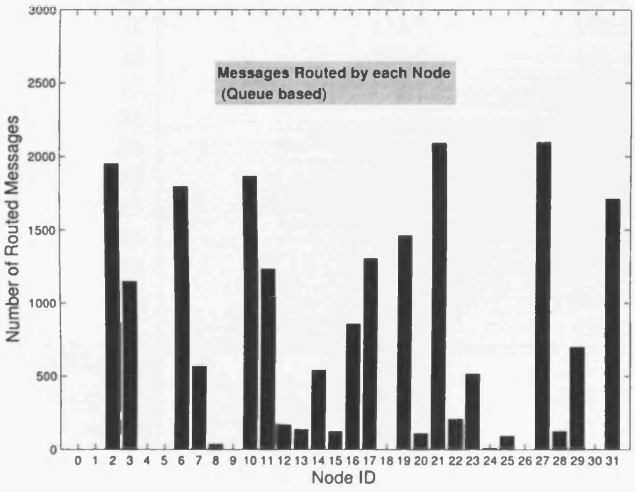



Figure 3.16: Message distribution with Queue Based Routing

In figure 3.15 node 21 routes messages to node 1 via node 2 only. Node 7 does not route any messages from 21. In figure 3.16 both node 2 and 7 node route messages effectively load balancing the network traffic. This is load balancing effect is archived but switching routes from one route (21-2-27-1) to another (21-1-11-31-10-1) as different routes are utilised. As queue length increases along the path 21-2-27-1

messages are routed along an alternate path 21-1-11-31-10-1. This is shown by an increase in the number of messages routed by these nodes.

Table 3.6 page 77 shows how the queue based routing outperforms shortest paths routing in every commander node position. It is most significant at positions 1,10,25,27.

| Commander Positions | Shortest Path Routing (ms) | Queue Routing (ms) |
|---|---|---|
| 0 | 10,096.20 | 7,219.20 |
| 1 | 2,867.60 | 186.4 |
| 2 | 134 | 131.3 |
| 3 | 553.4 | 537.4 |
| 4 | 443.5 | 177.1 |
| 5 | 3,043.80 | 2,955.20 |
| 6 | 112.8 | 109.8 |
| 7 | 127.7 | 122.6 |
| 8 | 149.3 | 146.2 |
| 9 | 156.3 | 150.3 |
| 10 | 2,262.60 | 161.1 |
| 11 | 130.8 | 124.3 |
| 12 | 156.9 | 148.3 |
| 13 | 693.8 | 168.4 |
| 14 | 3,037.60 | 2,949.10 |
| 15 | 3,096.60 | 3,006.40 |
| 16 | 2,052.50 | 1,992.60 |
| 17 | 544.1 | 528.2 |
| 18 | 585.2 | 568.1 |
| 19 | 531.5 | 516 |
| 20 | 138.9 | 130.7 |
| 21 | 113.7 | 113.2 |
| 22 | 425.2 | 156.4 |
| 23 | 147.4 | 141.6 |
| 24 | 170 | 166.5 |
| 25 | 3,148.10 | 218.2 |
| 26 | 3,104.90 | 3,014.50 |
| 27 | 1,783.00 | 189.4 |
| 28 | 3,048.00 | 2,959.20 |
| 29 | 3,009.10 | 2,921.40 |
| 30 | 140.5 | 134.7 |
| 31 | 140 | 132.6 |
| Minimum | 112.80 | 109.80 |
| Maximum | 10,096.20 | 7,219.20 |
| Average | 1,657.47 | 1,161.92 |

Table 3.6: Shortest Path Routing (figures are in ms unless otherwise stated)

### 3.7.4 Routing Test Case 2

When there are multiple links per face as shown in figure 3.17 performance of queue routing degrades. Queue routing does not respond quickly enough to temporal changes in link congestion. In order for queue routing be respond quickly enough then the weights associated with the links will need to be constantly update and therefore the routing tables would need to be constantly updated. The routing update could be made more frequent but this would be increase the routing overhead.

As queue routing is only updated every 0.25s, it can causes messages to be unnecessarily routed through longer paths to avoid congestion that no longer exists since the last routing update. Queue routing also does not take into account the link colour changes involved when using 2 links per face. An increase in colour switching increases delay as there is a minimum delay of 90ms per colour change. This occurs because if a message arrives on a blue timeslot but needs to be routed via a red timeslot to the next node, it will have to wait the 90ms for that red timeslot (see section 5.4.5).

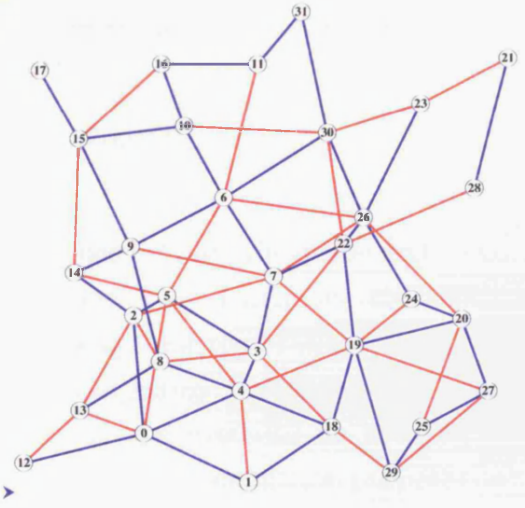



Figure 3.17: Test Graph with 2 Links Per Face

### 3.7.5 Adaptive Delay Routing

Routing using delay values as used in RIP [47] responds directly to delays along routes rather than queue lengths. The inclusion of colour changes is therefore inherent in the routing as it takes the 100ms delay in colour changes when distributing link state packets. Rather than implement a fully operational ad-hoc routing protocol such as AODV [74] or DSR [55] a very simple adaptive method has been included. This avoids having to fully implement a dynamic routing protocol. This

is only possible if all information is available at a central location and where the routing tables are built centrally. In practice the nodes will be independent and will require the full implementation of a dynamic network routing algorithm [8].

### 3.7.6 Adaptive Delay Routing Method

The routing in the simulation is based on a very simple RIP procedure but with the additional adaptive method:

```
Initialise shortest paths
    Repeat every 0.25s
        build Distance Vector Routing Tables
        attach delay values to distance tables
        compute new routes
        if better routes available
            update routing table
    Until end of simulation

If delays to the commander from congested routes drop below 500ms
   compute new routes
        If better routes available
            Update Routing Table
```

Notice that the only difference between the queue and Delay Routing is using delay along links rather than queues. The adaptive method keps track of congested routes and when message delays along these routes exceed 500ms new routes are calculated. This ensures that once the congestion has cleared, new messages will use these more direct routes. The computational overhead for tracking these congested routes is high but insignificant as they are for simulation purposes only. Full implementation would employ very lightweight, highly adaptive routing protocols and would almost eliminate this overhead [8].

The following three bar graphs show the messages routed by each node to the commander positioned at node 2 in the test graph shown in figure 3.17 page 78.

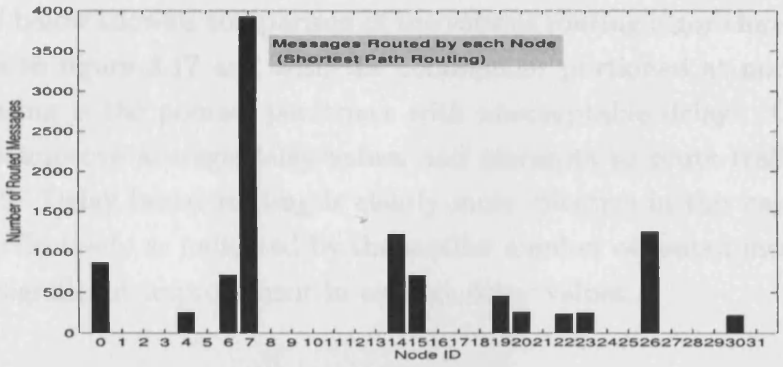



Figure 3.18: Message Distribution with Shortest path Routing

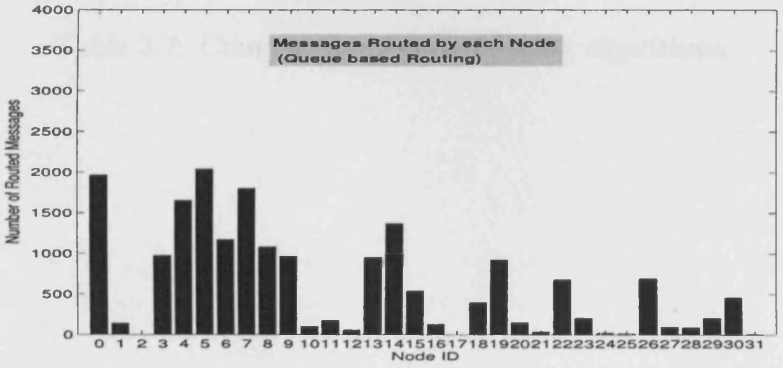



Figure 3.19: Message Distribution with Queue Routing

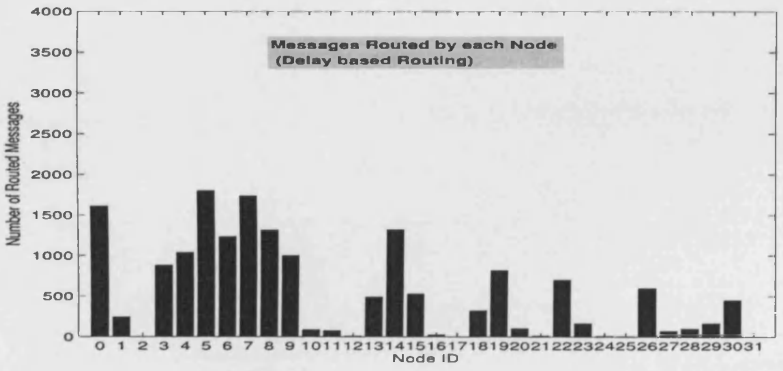



Figure 3.20: Message Distribution with Delay Routing

Table 3.7 below shows a comparison of the various routing algorithms for the test graph shown in figure 3.17 and with the commander portioned at node 2. Shortest path routing is the poorest performer with unacceptable delays. Queue based routing does improve average delay values and attempts to route traffic as shown in figure 3.19. Delay based routing is clearly more effective in this case. It routes traffic more effectively as indicated by the smaller number of routed messages. This results in a significant improvement in average delay values.

| | Shortest Paths | Queue based | Delay based |
|---|---|---|---|
| Total No. of messages routed (ms) | 10375 | 19062 | 16847 |
| Total No. messages received by commander (ms) | 7465 | 7465 | 7465 |
| Average msg delay (ms) | 14921.563 | 600.349 | 371.158 |

Table 3.7: Comparison of three routing algorithms

The results for delay based routing are consistently better than the queue based routing. Table 3.8 gives the average delay values (ms) for all command positions. Adaptive delay based routing outperforms queue based routing in all positions. Delay based routing minimum, maximum and average delay values were all lower. This trend is indicative of results throughout this thesis. Therefore adaptive delay routing is used throughout unless otherwise stated.

| Commander Position | Delay Based (ms) | Queue Based (ms) |
|---|---|---|
| 0 | 18,744.50 | 17477.6 |
| 1 | 3310 | 41647.1 |
| 2 | 357.2 | 594.6 |
| 3 | 291.5 | 711.5 |
| 4 | 302.5 | 572.5 |
| 5 | 337.60 | 504.1 |
| 6 | 291.8 | 623.8 |
| 7 | 273.7 | 740.7 |
| 8 | 349.5 | 634.6 |
| 9 | 368.5 | 532.4 |
| 10 | 414.2 | 940.7 |
| 11 | 12945.5 | 1465.3 |
| 12 | 14781.8 | 29633.8 |
| 13 | 451.1 | 10510.1 |
| 14 | 336.6 | 664.8 |
| 15 | 3956.5 | 30970.7 |
| 16 | 485.6 | 1746.9 |
| 17 | 32679.5 | 32617.5 |
| 18 | 350.4 | 738.5 |
| 19 | 283.5 | 398.6 |
| 20 | 408.7 | 635.8 |
| 21 | 32499.2 | 43247.1 |
| 22 | 307.5 | 510.3 |
| 23 | 16603.1 | 41991.1 |
| 24 | 380.9 | 697.7 |
| 25 | 543.5 | 714.9 |
| 26 | 317.4 | 651.3 |
| 27 | 521.1 | 1660.2 |
| 28 | 45017.6 | 50481.1 |
| 29 | 485.4 | 956.6 |
| 30 | 359.4 | 791.8 |
| 31 | 13031.8 | 17435.8 |
| Minimum | 273.70 | 398.60 |
| Maximum | 45,017.60 | 50,481.10 |
| Average | 6,305.85 | 10,421.86 |

Table 3.8: Queue vs Delay Based routing for all command positions. Average messages delay values are shown

## 3.8 Simulation Platform

Evaluating radar heuristic methods and simulation require terms such as *Quality*, *Performance* to be put into context. *Quality* can be the measure of the algorithms end result. *Performance* in terms of execution time, CPU cycles, etc, is dependent not only on the efficiency and design of the algorithm but also on external factors such as system specifications. Therefore *Performance* can be put into context by the system specification as shown in Table 3.9. This is in accordance with best practices discussed and set out in [87].

This thesis does not make direct speed efficiency comparisons with other algorithms from outside sources. However systems remain consistent to allow comparison of the various methods used throughout this work.

| GIS Server and SQL Server | |
|---|---|
| Motherboard | Intel Server Board SCB2 |
| Processor | Dual Intel Pentium 3 1.0Ghz |
| Processor Bus Speed | 133Mhz |
| Memory | 1Gb 4 X 256 ECC 133M DIMMS |
| Hard Disk 1 - System | Seagate Barracuda SCSI HD ST318418N |
| Hard Disk 2 and 3 - Microsoft SQL Store | Seagate Barracuda Ultra ATA 6 40810 40.0GB mirrored |
| Operation System | Windows 2000 Server Service Pack 2 - NTFS |
| Database (Server Application) | Microsoft SQL Server 2000 Service Pack 3 |
| GIS System (Workstation Application) | ESRI ArcView 8.1a |
| Network Adapter | Intel Pro/100 Ethernet Adapter |

| Design and Simulation Workstation | |
|---|---|
| Motherboard | Intel D850EMV2 Workstation Board |
| Processor | Intel Pentium 4 2.40 GHz |
| Processor Bus Speed | 533 MHz |
| Memory | 1 GB 800Mhz RAMBUS RAM 2 x 512 |
| Hard Disk 1 - System | Barracuda Ultra ATA 6 40810 40.0GB |
| Operation System | Windows* XP version 5.1 Service Pack 1 build 2600 |
| Network Adapter | Intel Pro/100 Ethernet Adapter |
| Custom Application | Built Using Microsoft Visual C++ Ver. 6 Service Pack 5 |
| External Libraries | Library of Efficient Data Types and Algorithms Ver 4.3.1 |

Table 3.9: System Specifications

## 3.9 System Inputs/Outputs

Figure 3.21 below shows the full system in the form of inputs to the system and generated outputs at each stage of the simulation and design.

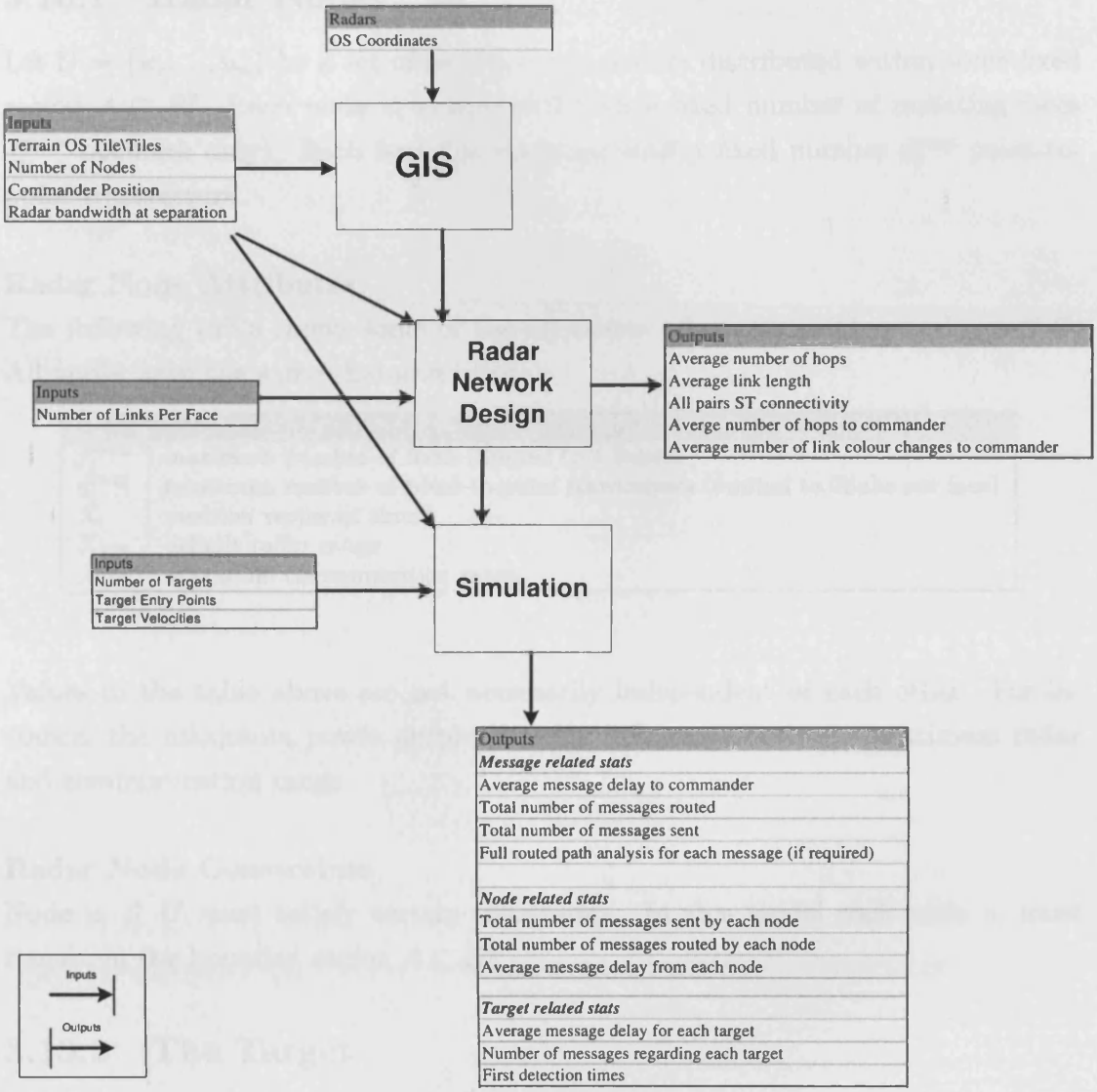



Figure 3.21: System overview inputs and outputs

## 3.10 Complete Model

The complete model of the netted radar system is defined more formally as follows:

### 3.10.1 Radar Nodes

Let $U = \{u_1, ..., u_n\}$ be a set of mobile radar devices distributed within some fixed region $A \subset R^2$. Each node $u_i$ is equipped with a fixed number of radiating faces $f_i^{max}$ (azimuth only). Each face $f$ is equipped with a fixed number $d_i^{max}$ point-to-point transceivers.

#### Radar Node Attributes

The following table shows some of the attributes associated with a node $u_i \subset U$. All nodes have the same characteristics.

| Node Attributes | |
|---|---|
| $f_i^{max}$ | maximum number of faces (limited to 4 faces) |
| $d_i^{max}$ | maximum number of point-to-point transceivers (limited to 2links per face) |
| $X_t$ | position vector at time t |
| $X_{RR}$ | default radar range |
| $X_{CR}$ | maximum communication range |

Values in the table above are not necessarily independent of each other. For instance, the maximum power output directly influences both the maximum radar and communication range.

#### Radar Node Constraints

Node $u_i \subset U$ must satisfy certain constraints. In this model each node $u_i$ must remain in the bounded region $A \subset R^2$.

### 3.10.2 The Target

A target $t'$ travels across a bounded region $A \subset R^2$ with a velocity $v'$. For any given region $A$, there can be a maximum of $t_{max}$ targets.

### 3.10.3 Network Traffic

A radar network is built primarily to support the communication of traffic (in the form of packets of information) between a node $u_i$ and the commander node $u_c$ where $u_c$ is responsible for aggregating all information and making decisions based on this information.

This model contains only traffic relating directly to target detections and tracking. As we are looking primarily at the design performance of a new radar concept,

control traffic is considered to be communicated instantaneously. Control traffic can consist of synchronising, maintaining and routing procedures for the radar network.

**A packet based radar traffic model**

A packet based traffic model can be represented as the transfer of discrete unit of information in the form of messages (as in section 3.2.3) between two nodes $u_i$ and a commander node $u_c$. In this model, a message $m$ is only sent after a target is within range of a radar node.

**Message creation event**

If at time $t_1$, the distance between a radar node $u_i$ and target $t'$, $d(t', u_i) < X_{RR}$ radar mode is changed to surveillance mode which is effectively $1.5 \times X_{CR}$ . At time $t_2 = t_1 + 2seconds$ a message $m'$ regarding $t_1$ is generated at time $t_2$ which requires transmission from $u_i$ to $u_c$ via the radar network. While $d(t', u_i) < 1.5 \times X_{CR}$ a message $m'$ is sent to $u_c$ at one second intervals. At time $t_3 = t_2 + 2seconds$ radar mode is changed to tracking mode. Now a message $m'$ is sent to $u_c$ at half second intervals. All nodes $u_i \subset U$ operate in this manner.

### 3.10.4 Radar Links

Let $u_i, u_j \subset U$, let $(u_i, u_j)$ denote a link from $u_i$ to $u_j$ and let $\mathcal{N}_i$ be a set of nodes $u_j \subset U$ with the property that $u_i$ is able to support a link $u_j$. $\mathcal{N}_i$ is the set of potential neighbours of $u_i$ subject to conditions on the link and constraints on the nodes.

**Radar Link Constraints**

The potential neighbours $\mathcal{N}_i$ are those that are within the maximum communication range $X_{CR}$ of the radar node. Therefore all nodes $u_j$ are located within a disc of radius $X_{CR}$.

In this model the capacity is limited only by distance. The separation bandwidth $B_{sp}$ for this radar application is modeled based on the data rate at a separation distance of 1km and is as follows:

$$B(u_i, u_j) = \frac{B_{sp}}{d(u_i, u_j)^2}$$

where $d(u_i, u_j)$ is the distance between nodes $u_i$ and $u_j$. In this thesis the reference data rate $B_{sp}$ has been obtained through experiment by Qinetiq Malvern. The $1/d^2$ term is related to the concepts described in section 1.

Each node $u_i$ has limited number of links available per face. For link $(u_i, u_j)$, face $f_{ui}$ directed at $u_j$ must have allocated less than the total number of allowed links per face $d_i^{max}$. Similarly face $f_{uj}$ directed at $u_i$ must have allocated less than $d_i^{max}$ links. Therefore nodes $u_j$ that have $d_i^{max}$ allocated links on faces that are directed at $u_i$ are not contained in $\mathcal{N}_i$.

### 3.10.5 Radar Network

A complete radar network is made up of a set of radar nodes $U$ where $U$ is the total number of radar nodes in the network. The radar nodes are connected by a set of edges $E$ connecting nodes $u_i, u_j \subset U$. The radar network is therefore represented by a graph with a set of nodes $U$ and edges $E$, $G(U, E)$. Each network has a commander node represented by $v_C$.

**Radar network design objective**

The key objective in constructing a radar network is to minimise the average delay of all messages $M$ sent to the commander $v_c$ regardless of which radar node is designated the commander position. A network design is considered acceptable for a particular command position $v_c$ if the average delay of all messages $M$ is less than 500ms. All nodes in $U$ must to be connected in order to give the best available radar coverage.

CHAPTER

# FOUR

## Radar Network Global Optimisation

This thesis does not attempt to compare the different methods currently used in state-of-the-art multi-objective optimisation methods. During the early stages of the project, SPEA was chosen because of its proven performance with respect to other multi-objective optimisation methods. (Zitzler [27] gives a detailed comparison of the state-of-the-art optimisation in late 1999).

SPEA has not been developed for network design problems to date. SPEA does provide a framework whereby the algorithm can solve some difficult optimisation problems such as a software synthesis and the Knapsack problem [26] The straightforward nature of the algorithm lends itself to being developed and tailored to our problem. The aim is not to find an optimal solution but a good solution in a reasonable amount of time. With Genetic algorithms being used in more diverse and difficult areas [35] algorithm efficiency becomes more important. As the complexity of the problems, search space, and evaluation procedures increase, the algorithms also need to be able to manage and adapt to these more complex search spaces to ensure the good solutions can be reached in reasonable time.

## 4.1 Metrics Used

As far as is known this project is the first to have considered netted radar in communications network terms. Therefore there are no established radar specific metrics that can be used to assess the quality of the radar network. However there are metrics that can be applied to these networks which produce good network performance.

1. Number of Hops
2. Link length (bandwidth)
3. ST connectivity

These simple metrics only give an indication of the quality of the network from a topology point of view. In order to evaluate performance of the networks, the operations of the network under simulation will be evaluated.

***Number of Hops*** Number of hops is the measure of the path cost from a source node $s$ to a target node $t$. Fewer hops in networks tend to be better as they consume less network resources in total. When a node is not able to establish a direct link to a required node, then there is no option but to use a multiple hop route. Figure 4.1 illustrates this situation. Node A can only send to node C through Node B as a direct link cannot be established.

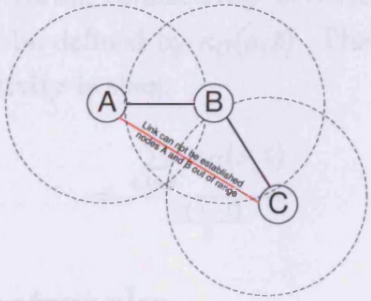



Figure 4.1: Multi-Hop

The average number of hops is calculated using the total number of minimum hops between each pair of nodes. Minimum hop paths $min(p)$ are calculated using shortest paths as stated in section 3.7.1 whereby each link is given a weight of 1. The weight of each minimum path $min(p) : p(u \rightarrow v)$ for each pair of nodes $u, v \in V$ is summed and then divided by the total number of node pairs where $n$ is the number of nodes. Therefore:

$$\text{Average Number of Hops} = \frac{\sum_{u,v \in V} min(p) : p(u \rightarrow v)}{\frac{n(n-1)}{2}}$$

***Average Link Length*** The average link length calculation is simply the sum of all link lengths divided by the number of links. For a network of $V$ nodes and $E$ links, the $LinkLength(e)$ is the link length of a link $e \in E$ where the length of a link $e$ is the distance between adjacent nodes of $e$, $u$ and $v$ where $u, v \in V$ .

$$AverageLinkLength = \frac{\sum_{e \in E} \text{Length Link}(e)}{|E|}$$

where $|E|$ is the number of links in $E$.

***All-pairs ST Vertex Connectivity*** ST connectivity and more generally vertex connectivity is characterized by Menger's Theorem. Menger's (1927) Theorem: Let $G = (V, E)$ be a graph and $A, B \subseteq V$. Then the minimum number of vertices separating $A$ from $B$ in $G$ is equal to the maximum number of vertex disjoint $A - B$ paths in $G$. The interested reader is referred to [23] for a complete analysis of Menger's theorem. If the number of vertex disjoint paths is known, then the vertex connectivity between the nodes $A$ and $B$ is known. If, for instance, there are three vertex disjoint paths between $A$ and $B$, then the minimum number of vertices that need to be removed to separate $A$ from $B$ is three.

This thesis utilises Menger's theorem to find the vertex connectivity between all pairs of nodes. Let the vertex connectivity between a pair of nodes $s$ and $t$ in $G = (V, E)$ where $s, t \in V$, be defined by $\kappa_G(s, t)$. The number of nodes in $V$ is $n$. The average vertex connectivity is then:

$$= \frac{\sum_{s,t \in E} \kappa_G(s,t)}{\frac{n(n-1)}{2}}$$

## 4.2 Encoding networks

The SPEA provides the general framework to solve optimisation problems. The encoding of these problems is problem specific. Here the networks are encoded using a binary string where each bit represents a link. As the capacity of the link is dependent on link length itself, it does not have to be encoded directly. This very simple method of encoding networks is particularly suited to links with only two values i.e. on or off, 1 or 0 (Michalewicz [65] illustrates the use of binary strings to encode the travelling salesman problem). The string encoding shown in figure 4.2 is very similar.

Encoding the networks using a binary string with bits representing edges has obvious scaling problems as networks increase in size. The advantages are that comparisons can be done extremely fast using binary subtraction. An example string is shown below figure 4.2 for a 5 node network. Each 1 value in the binary string represents a valid link between 2 nodes.

| Link Number | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| Link- Node A | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 2 | 2 | 3 |
| to Node B | 1 | 2 | 3 | 4 | 2 | 3 | 4 | 3 | 4 | 4 |
| Binary String Value | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |

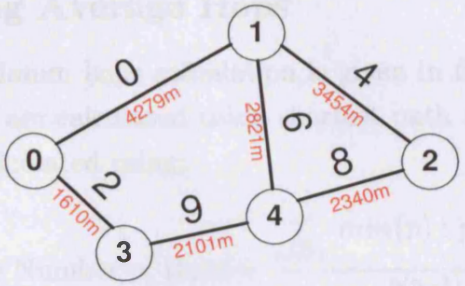



Figure 4.2: Network Encoding (red figures indicate distance between nodes)

The bit string length/chromosome length is dependant on the number of nodes $n$, where

$$\text{Chromosome Length} = \frac{n(n-1)}{2}$$

giving 10 in this particular instance.

From section 4.1, average link length of the network shown in figure 4.2 is

$$AverageLinkLength = \frac{\sum_{e \in E} \text{Length Link}(e)}{|E|}$$

$$AverageLinkLength = \frac{4279 + 2821 + 3454 + 1610 + 2101 + 2340}{6} = 2767.5m$$

### 4.2.1 Calculating Average Hops

An example of the minimum hops calculation is given in figure 4.3. Both network 1 and network 2 tables are calculated using shortest path algorithms and then the average hops is then calculated using:

$$\text{Average Number of Hops} = \frac{\sum_{u,v \in V} min(p) : p(u \rightarrow v)}{\frac{n(n-1)}{2}}$$

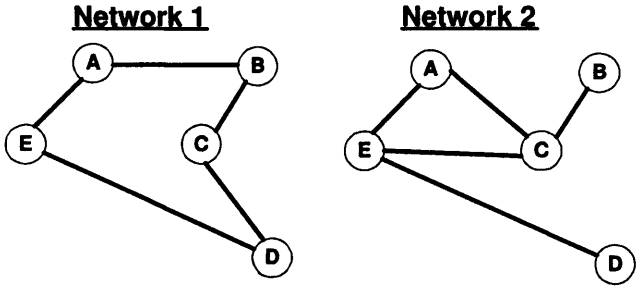



Figure 4.3: Multi-Hop

**Network 1**

| | A | B | C | D | E | Total |
|---|---|---|---|---|---|---|
| **A** | | 1 | 2 | 2 | 1 | 6 |
| **B** | 1 | | 1 | 2 | 2 | 6 |
| **C** | 2 | 1 | | 1 | 2 | 6 |
| **D** | 2 | 2 | 1 | | 1 | 6 |
| **E** | 1 | 2 | 2 | 1 | | 6 |
| | | | | | | 30 |

**Network 2**

| | A | B | C | D | E | Total |
|---|---|---|---|---|---|---|
| **A** | | 2 | 1 | 2 | 1 | 6 |
| **B** | 2 | | 1 | 3 | 2 | 8 |
| **C** | 1 | 1 | | 2 | 1 | 5 |
| **D** | 2 | 3 | 2 | | 1 | 8 |
| **E** | 1 | 2 | 1 | 1 | | 5 |
| | | | | | | 32 |

Total number of hops for network 1 is therefore 30/20 = 1.5 hops on average from any node s to any node t. In network 2 the average number of hops is 32/20 = 1.6.

### 4.2.2 Calculating connectivity

Calculating the All-pairs ST vertex connectivity requires calculating the maximum flow between all pairs of nodes. The All-pairs ST vertex connectivity is calculated as described in 4.1. The full details of the "pre-flow push" algorithm used to calculate the flows is given in [97]. Implementation details and analysis are given in [16].

## 4.3 The SPEA Algorithm

The SPEA algorithm uses the notion of pareto dominance (section 2.3.1) together with a new concept of *strength* to drive the pareto front (section 2.3.1) forward. Consider the example shown in figure 4.4. It shows a set of ten solutions. The *external set* contains members of the pareto front (all non-dominated solutions). SPEA uses the the concept of an external set to represent the best solutions. This external set is calculated based on the number of solutions that each external set member dominates.

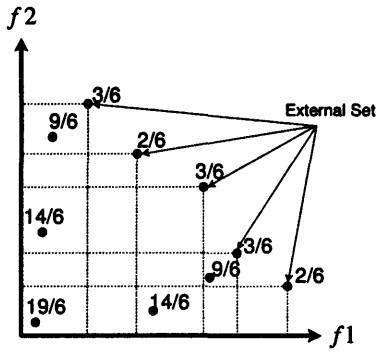



Figure 4.4: SPEA example

The SPEA overview for Radar networks proceeds as follows:

1. ***(Initialisation)*** An initial population $P_t$ of size $N$ of binary encoded radar networks is generated. This involves randomly selecting initial links. Repair operators[1] are implemented at this early stage to ensure that the radar networks are valid.

2. ***(Update External Set)*** The external set $\overline{P_t}$ is re-evaluated after each new generation. This requires new nondominated solutions to be added and now dominated solutions from the external set to be removed. There is also a clustering algorithm used to ensure that the external set is reduced to ensure that it does not grow above its maximum number of solutions.

3. ***(Assign fitness values)*** This is the *strength* element of the algorithm. Strengths $S$ for each member $i$, $S(i)$ of the external set $\overline{P_t}$ are calculated based on the number of solutions $j$ in $P_t$ it dominates and is calculated as follows:

$$S(i) = \frac{|j|j \in P_t \wedge m(i) \succeq m(j)|}{N+1}$$

[1]Repair operators are methods of removing infeasible elements from a solution by either removing them completely or using a heuristic method to correct the solution

Here the number of total number solutions which vector i dominates is divided by the population size $N+1$. This figure represents the external set strength. The fitness of the individuals j i.e. the dominated individual is calculated by

$$S(j) = 1 + \sum_{i \in \overline{P_t} \wedge m(i) \succeq m(j)} S(i)$$

The strength of the dominated individuals is calculated by the sum of all the dominating individuals strengths S(i) and adding 1.

*A complete example of the fitness calculation is shown in figure 4.4.*

4. ***(Selection)*** A new population is generated by choosing two solutions at random from either the external set or population and making them compete in binary tournaments. A binary tournament is the random selection of two individuals $a$ and $b$ from either the external set or population. These two individuals are compared with respect to strength and if $S(a) > S(b)$ then $a$ is put forward the the next generation else $b$ goes through. In other words, the strongest individual goes through to the next generation and the loser is removed.

5. ***(Crossover,Mutation and Repair)*** The individuals then randomly undergo crossover and mutation based on crossover and mutations probabilities. The resulting solutions are repaired to ensure they form valid radar networks (this is discussed later in sections 4.8.1).

6. ***(Termination)***If the total number of generations is complete then the external set represents the final pareto front. Otherwise the algorithm is run from the *Update External Set* phase.

The SPEA algorithm described is almost identical to the original framework set out in [26]. Apart from the custom encoding of the radar networks and the repair operators, which are radar specific, the framework remains almost unchanged. It is also important to recognise that the actual values of $f1$ and $f2$ do not affect the operation of the algorithm as the fitness/strengh is calculated based entirely on dominance. This allows criteria to be interchanged without altering the algorithm itself. It is also important to note that there is no cost function. The algorithm is driven solely by pareto dominance.

After completion of the SPEA radar network design, a set of candidate network designs for a given problem is produced. These candidate network designs have to be coloured using the procedure outlined in section 3.6.3. This post-algorithm design step takes place independently of the radar network SPEA but uses the candidate

designs as input. The colouring is a vital step as it dictates the correct timings of the communication links.

The radar networks are encoded as set out in Section 4.2. The network criteria that can be used in conjunction with this algorithm is described in section 4.1. A complete description of the algorithm is shown on the following page.



**Algorithm 4.3.1:** SPEA ALGORITHM FOR RADAR NETWORKS()

*variables*
$N$ Population Size
$\overline{N}$ Maximum Size of External Set
$T$ Maximum number of generations
$p_c$ Crossover probability
$p_m$ Mutation probability
$p_t$ Set of decision vectors in population at time t
$\overline{p_t}$ Set of decision vectors in external set at time t
$p_{temp}$ Tempory population of maximum size N to store next generation of solutions
$A$ Final nondominated set

/* ***Initalisation*** - */
**for** $i \leftarrow 0$ **to** $N$
**do** Generate random BERN
$p_t \leftarrow p_t + BERN$
$\overline{p_0} \leftarrow 0$ /* Set external set to empty 0 */

**for** $i \leftarrow 0$ **to** $T$
/* ***Update External Set:*** */
Copy nondominated members $\overline{p_t} \leftarrow m(p_t)$
Remove dominated individuals from $\overline{p_t}$
Reduce number of decision vectors in external set $\overline{p_t}$ to $\overline{N}$

/* ***Assign population and external set strengths:*** */
Calculate fitness based on SPEA strengths procedure

External set fitness: $S(i) = \frac{|j|j \in P_t \wedge m(i) \succeq m(j)|}{N+1}$

Population set fitness: $S(j) = 1 + \sum_{i \in \overline{P_t} \wedge m(i) \succeq m(j)} S(i)$

/* ***Selection:*** */
**for** $i \leftarrow 0$ **to** $N$
Select two individuals $i, j \in p_t + \overline{p_t}$
if $F(i) < F(j)$ $p_{temp} = p_{temp}$ + i else $p_{temp} = p_{temp}$+ j

/* ***Crossover and Mutation:*** */
Crossover solutions from $p_{temp}$ with probability $p_c$
Mutate solutions from $p_{temp}$ with probability $p_m$

/* ***Repair population:*** */
Repair all solutions based on radar network criteria

/* ***Re-Initialize Population:*** */
Copy $p_{temp}$ to $p_t$

**end for**

/* ***Extract final nondominated set:*** */
$A = \overline{p_t}$

***Functions:***
*BERN* Binary Encoded Radar Network

## 4.4 Designing Centrally Controlled Networks

During the initial stages of the project, the research surrounding the radars themselves indicated that there might be up to 200 nodes in a single network. More recent research in the area [84] suggests that networks exceeding 32 nodes are unlikely.

The SPEA can optimise two parameters simultaneously as described in section 4.3. The SPEA attempts to optimise the graphs based on the parameters used. Its uses conventional crossover and mutation, the difference being that the external set and population size have to be set in order to obtain the correct selection pressure and diversity. During the project the following parameters were used for the majority of tests

- Crossover 70%
- Mutation 0.01
- Population Size 40
- External Set 25

The crossover and mutation values are based on results and recommendations of [20]. External set and population sizes are recommended in [27].

Initially the SPEA algorithm was run with average number of hops and average link length. Algorithms described in section 2.2 all use some element of capacity and some element of hops (such as MST and CMST). These criteria can also be calculated quickly and therefore it seems appropriate to test the radar networks with the SPEA algorithm.

## 4.5 Complete Graph

The graph below shows the complete graph and node configuration.

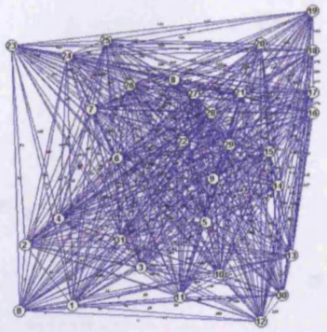


Figure 4.5: Complete Graph

The graph is encoded as a 496 bit string (see section 4.2) and represents the complete graph. This represents the total search space without any constraints.

## 4.6 Search Space

Having incorporated the network encoding and parameters onto the SPEA framework, the algorithm can be considered an unconstrained optimisation method. For the complete graph of 32 nodes, the search space is extremely large. For a network with $n$ nodes and $p$ discrete links sizes[2]:

$$\text{Number of possible solutions} = p^{\frac{n(n-1)}{2}} = 2.04 \times 10^{149}$$

## 4.7 Exhaustive Search

As SPEA has not to date been used as a network design tool, its relative or true performance as a network design tool has not been established. In order to test the effectiveness, a benchmark needs to be established. The algorithm can then be tested based on the proximity of its solutions to this bench mark. An exhaustive search of a set of nodes will provide the optimal graphs as a benchmark. Figure 4.6 below shows an exhaustive search of an 8 node network and SPEA search of the same 8 node network. The SPEA was run for 5000 generations.

[2]In radar networks $p = 2$, 0 and 1, 0 implying no link present and 1, link is present.

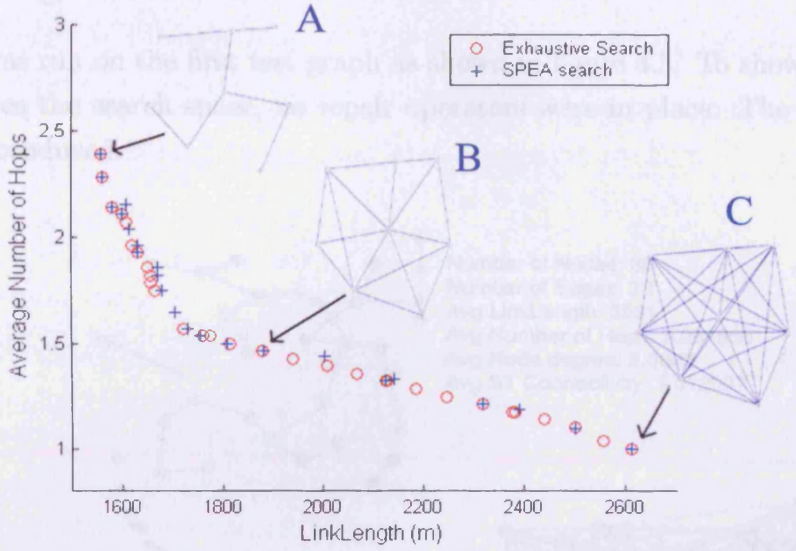



Figure 4.6: Graph of exhaustive search and SPEA search

The exhaustive search shown in Figure 4.6 details 37 optimal graphs for networks designed with SPEA setup for minimised average number of hops and average link length. The SPEA performs well, matching or closely matching the optimal network design. The SPEA finds both the minimum spanning tree (shown in figure 4.6 as A) and the fully connected graph (shown in figure 4.6 as C). Obviously graph C has too high a node degree for radar networks. However it is interesting to note that graph B represents an optimal design as it has used the maximum number of 2 links per face and is a valid radar network. A valid network has radar nodes that do not exceed the maximum number of links per face and all links between nodes do not exceed the radar communication range of the radar nodes.

Although an 8 node graph does not approach the network sizes dealt with in this project, the extremely large search space make larger exhaustive searches prohibitively computationally expensive.

$$\text{Number of possible solutions for 8 Node Graph} = p^{\frac{8(8-1)}{2}} = 2.68 \times 10^8$$
$$\text{Number of possible solutions for 9 Node Graph} = p^{\frac{9(9-1)}{2}} = 6.87 \times 10^{10}$$

The single exhaustive search as shown in figure 4.6 takes over 6 weeks with the SPEA design taking 16 seconds. An exhaustive search of a 9 node network would expect to take 2 years! However, this test does indicate that the SPEA developed in this thesis has the capability to obtain optimal or near optimal design.

## 4.8 Designs and Constraints

The SPEA was run on the first test graph as shown in figure 4.5. To show how the SPEA explores the search space, no repair operators were in place. The following results were produced.

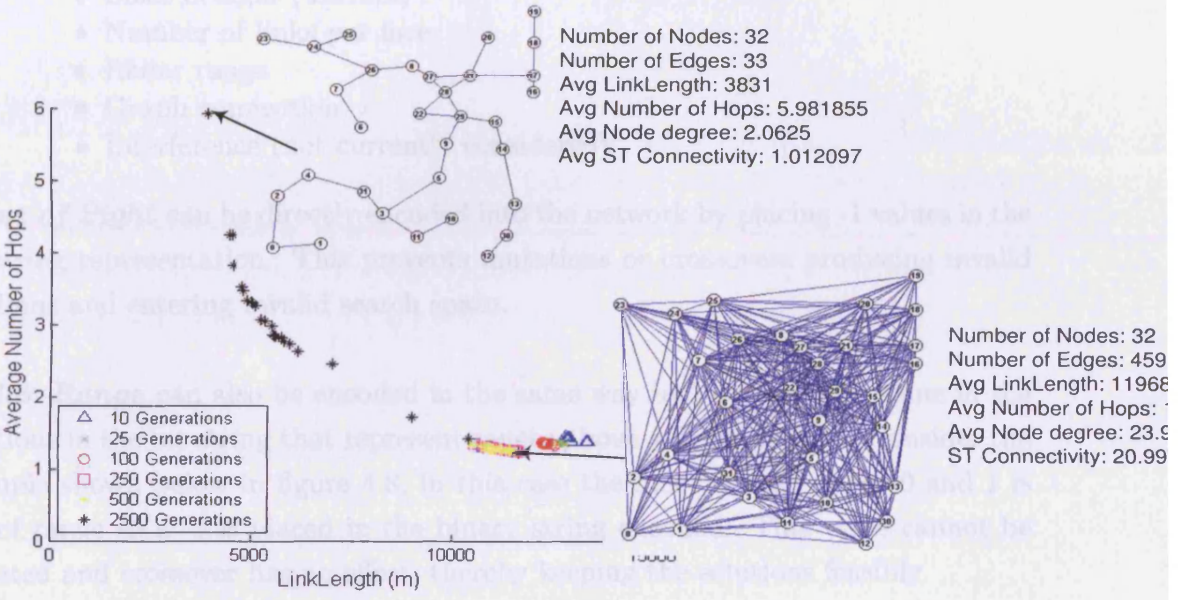



Figure 4.7: Initial SPEA network optimisation

The algorithm starts by selecting 1 link per face for each node to ensure that the network is connected. Having non-connected graphs produces infeasible solutions so to ensure that this does not occur, the starting graphs are highly connected.

During the early generations the solutions drive towards the Pareto front. After approximately 250 generations the external set lies on, or very close to the Pareto front. Further generations explore the breadth of the Pareto front. The large number of possible Pareto optimal solutions causes an uneven distribution along the Pareto front.

There is obviously a difficulty in visualizing the performance of the multi-objective optimisation. This is mainly because single solutions cannot be followed. Figure 4.7 shows two solutions at the extremes of the Pareto front. The SPEA shows adequate performance as both solutions shown are very close to the optimal solutions after 2500 generations (a minimum spanning tree and the complete graph). Fine tuning the population size, external set size, crossover and mutation can improve the performance.

### 4.8.1 Network Constraints and Reducing Network Complexity

The constraints required for this problem come from the hardware constraints of the radars themselves and their environment. The constraints are therefore as follows:

- Lines of sight (Terrain)
- Number of links per face
- Radar range
- Graph connection
- Interference (not currently considered)

***Lines of Sight*** can be directly encoded into the network by placing -1 values in the bit string representation. This prevents mutations or crossovers producing invalid solutions and entering invalid search space.

***Radar Range*** can also be encoded in the same way by inserting a -1 value in the positions in the bit string that represent ranges above a defined range. Consider the example shown below in figure 4.8, in this case the link between nodes 0 and 1 is out of range so a -1 is placed in the binary string encoded. This value cannot be mutated and crossover has no effect, thereby keeping the solutions feasible.

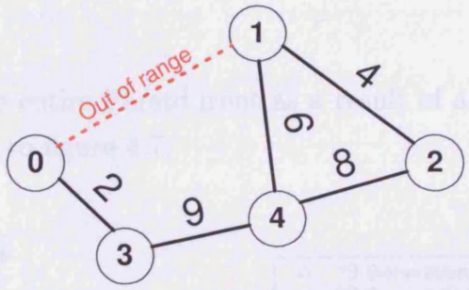



Figure 4.8: Network Encoding

| Link Number | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| Link- Node A | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 2 | 2 | 3 |
| to Node B | 1 | 2 | 3 | 4 | 2 | 3 | 4 | 3 | 4 | 4 |
| Binary String Value | -1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |

In the example 32 node network shown in figure 4.5 page 98, imposing a restriction on link length of 7500 metres reduces the search space to $3.14 \times 10^{88}$.

***Number of Links per Face*** The number of links per face cannot be encoded into the network. This can only take the form of a repair operator. As the repair operator is fairly complicated it is expected not only to be resource intensive but also to slow the optimisation. This constraint however is fundamental to the operation

of the network. The repair operator is outlined below where *ValidNumberOfLinks* is the number of links allowed on each face( in this thesis it is strictly limited to 1 or 2 links per face).

```
For Each Node
    For Each Face
        While{NumberLinksOnFace > ValidNumberOfLinks}
            Remove Random Link
    EndFor
EndFor
```

***Graph Connection*** This is a check constraint that tests the network to ensure that it remains connected. If the network is disconnected then a link is randomly added between disconnected components until the network becomes connected. The original technique was first developed by Hopcroft and Tarjan [98].

***Interference*** Interference is not currently considered in the model although the model has been built to include relevant ITU interference models. The GIS components developed for use in the simulator(section 3.4) allow any relevant propagation/attenuation equations to be incorporated into the GIS component. Models particular to the deployment area of the radar system can then be used if and when required.

Figure 4.9 shows the entire Pareto front as a result of an unconstrained optimisation process identical to figure 4.7.

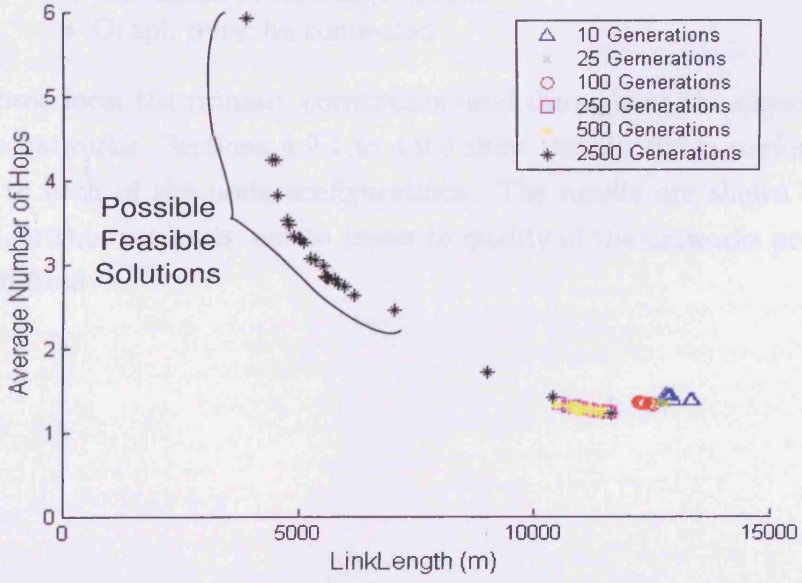



Figure 4.9: Feasible SPEA Network Designs

The feasible region in figure 4.9 indicates how solutions in this region *could* contain solutions that are feasible as radar networks. Clearly networks with links over 7500m are not viable and are therefore outside the feasible region. Networks in the feasible region must also conform to the links per face rule, all nodes must be connected and all links have to be less than maximum radar communication range. Therefore the network constraints/repair operators have to be applied to ensure that the solutions in this region are feasible.

## 4.9 Constrained SPEA

In order to obtain feasible results from the SPEA, constraints need to be imposed. The SPEA was run on thirty test networks:

- Ten 32 nodes networks
- Ten 64 node networks
- Ten 128 node networks

Each test network represents a non-random node placement. All placements were placed non-randomly in order to maintain a high level of coverage between 90%-100%. This represents thirty unique node configurations. This allows the SPEA algorithm to be tested on a wide range of node configurations and node sizes.

Constraints are applied to the three sets of Networks as follows

- Lines of sight are valid
- Number of links per face must be less than 2
- Maximum radar range 7500m
- Graph must be connected

These form the primary constraints used throughout the algorithm to maintain viable networks. Sections 4.9.1 to 4.9.3 show the algorithm performance when applied to each of the node configurations. The results are shown to illustrate how the algorithm proceeds, not to assess to quality of the networks produced along the Pareto front.

### 4.9.1 Constrained SPEA Test Results for 32 Node Networks

The following figures 4.10 and 4.11 show the incremental SPEA results for each node configuration. For these graphs terrain has not been considered. Running times for the test networks vary from 15-20 seconds for 1000 SPEA generations. This is consistent with rapid network design.

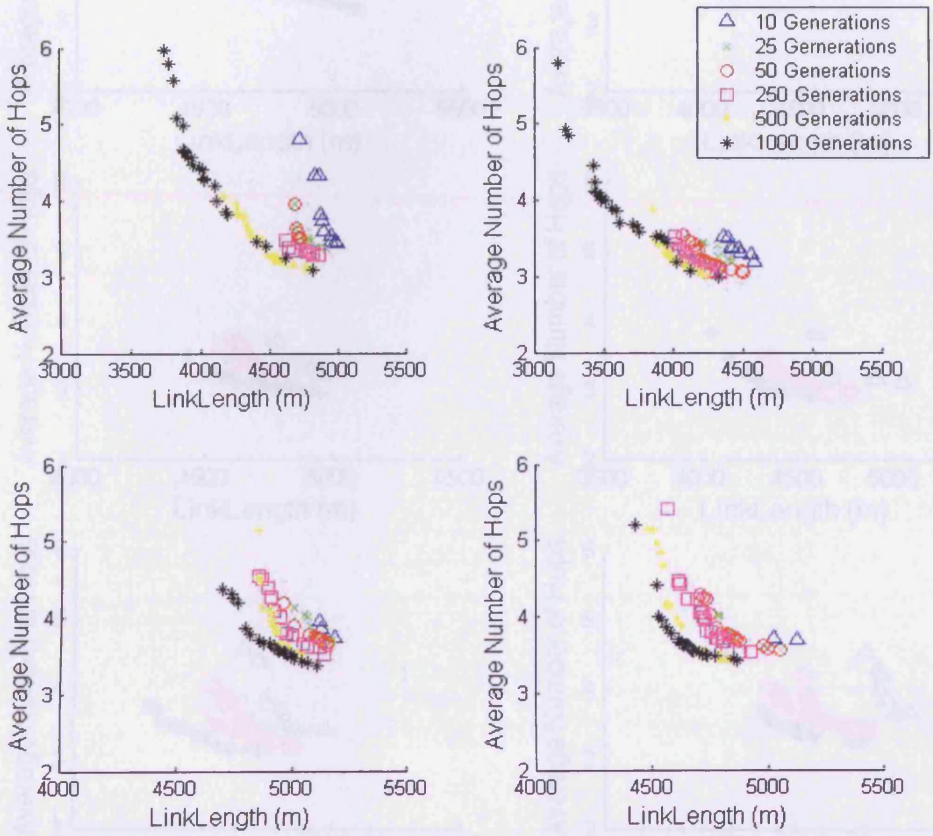



Figure 4.10: SPEA Results 32 nodes with constraints Tests 1-4

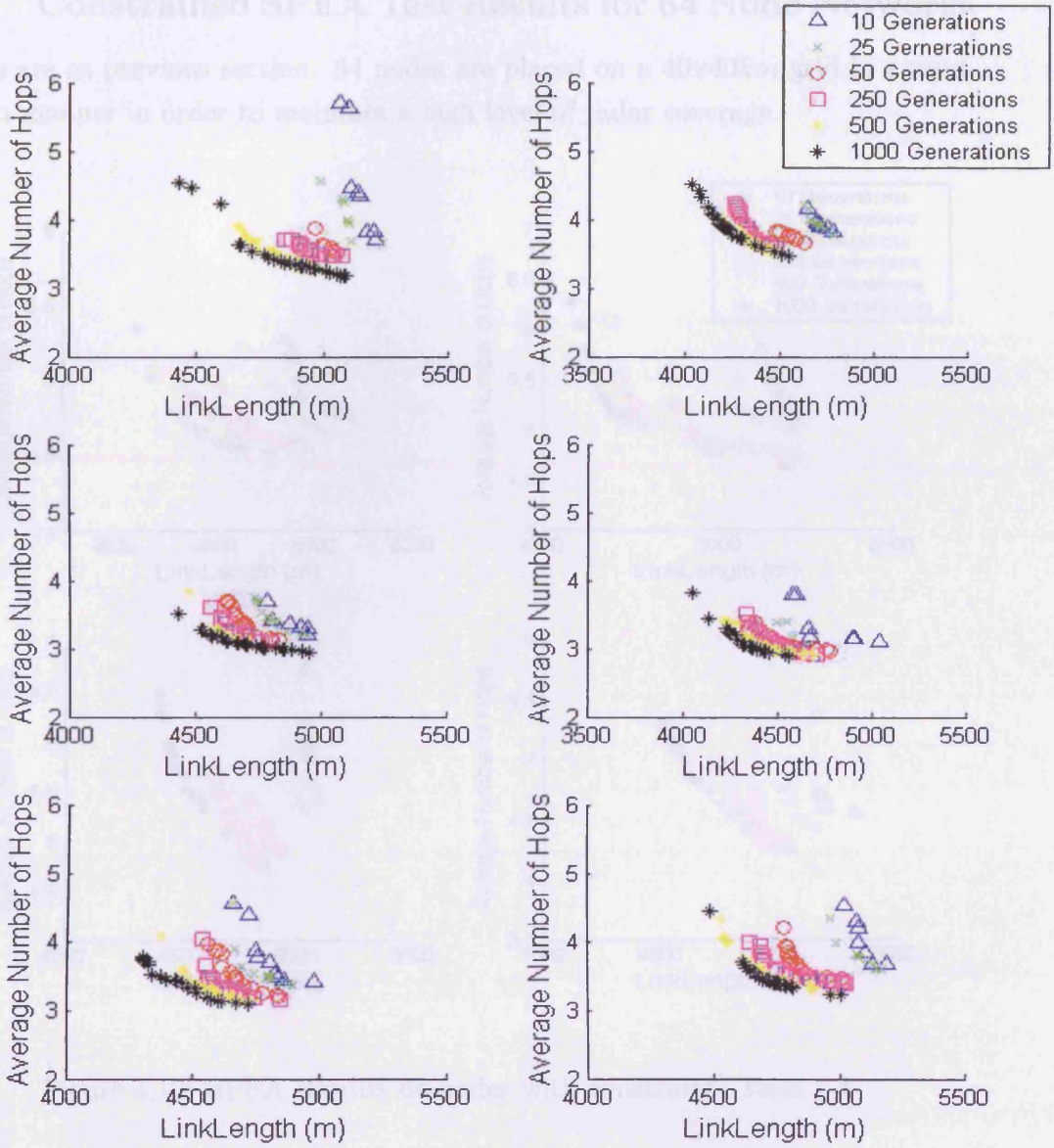



Figure 4.11: SPEA Results 32 nodes with constraints Tests 5-10

### 4.9.2 Constrained SPEA Test Results for 64 Node Networks

Results are as previous section. 64 nodes are placed on a $40x40km$ grid in a semi-random manner in order to maintain a high level of radar coverage.

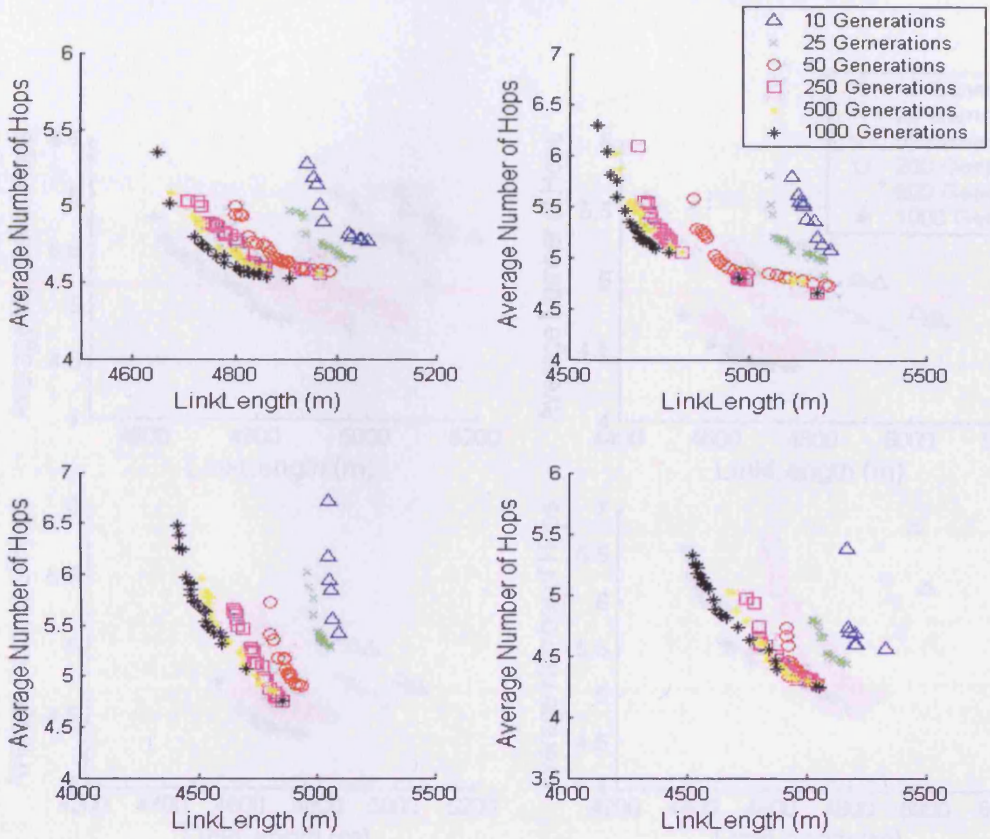



Figure 4.12: SPEA Results 64 nodes with constraints Tests 1-4

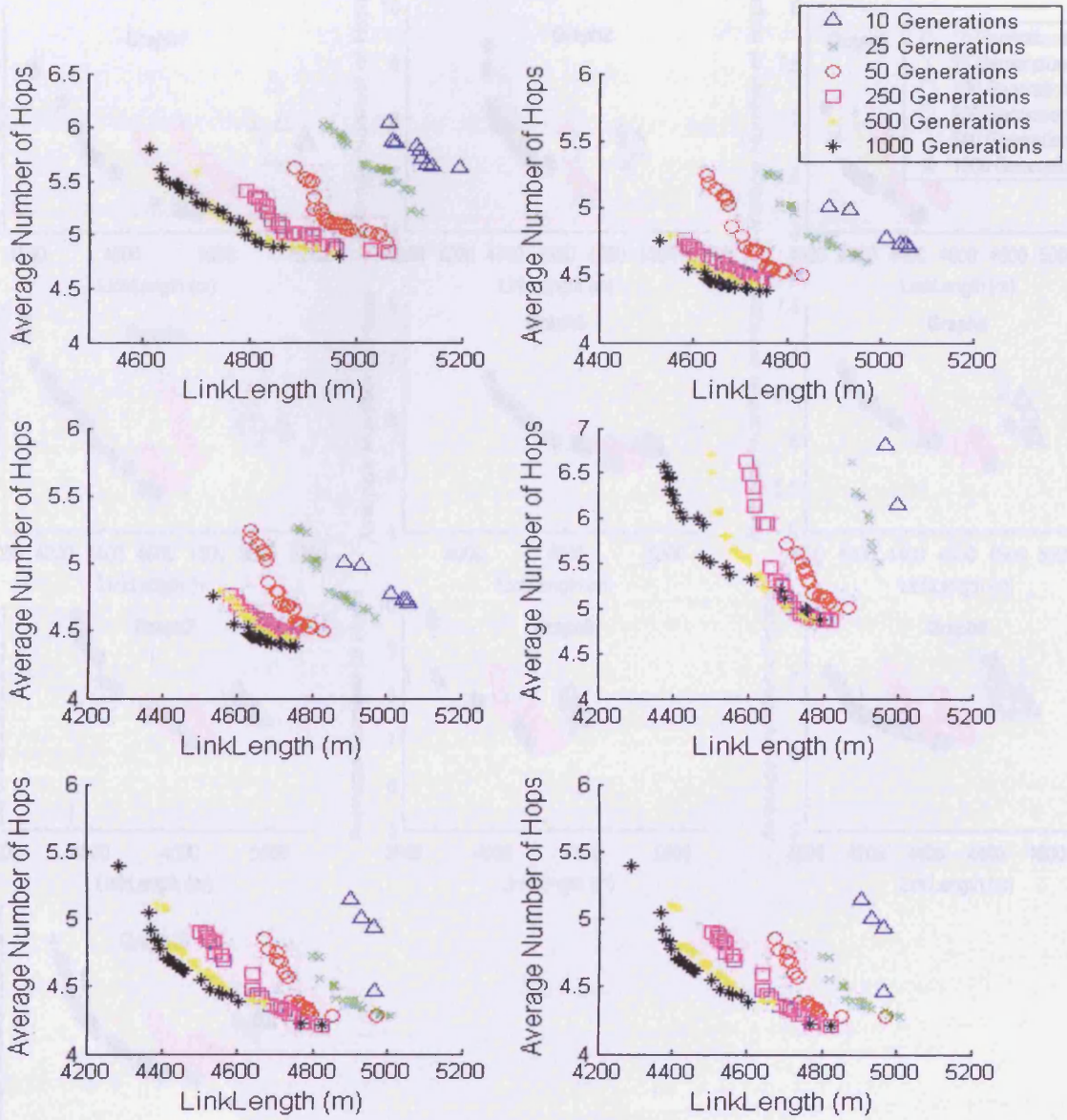



Figure 4.13: SPEA Results 64 nodes with constraints Tests 5-10

### 4.9.3 Constrained SPEA Test Results for 128 Node Networks

Results are as previous sections. 128 Nodes are placed on a 56x56$km$ grid in a semi-random manner in order to maintain a high level of radar coverage.

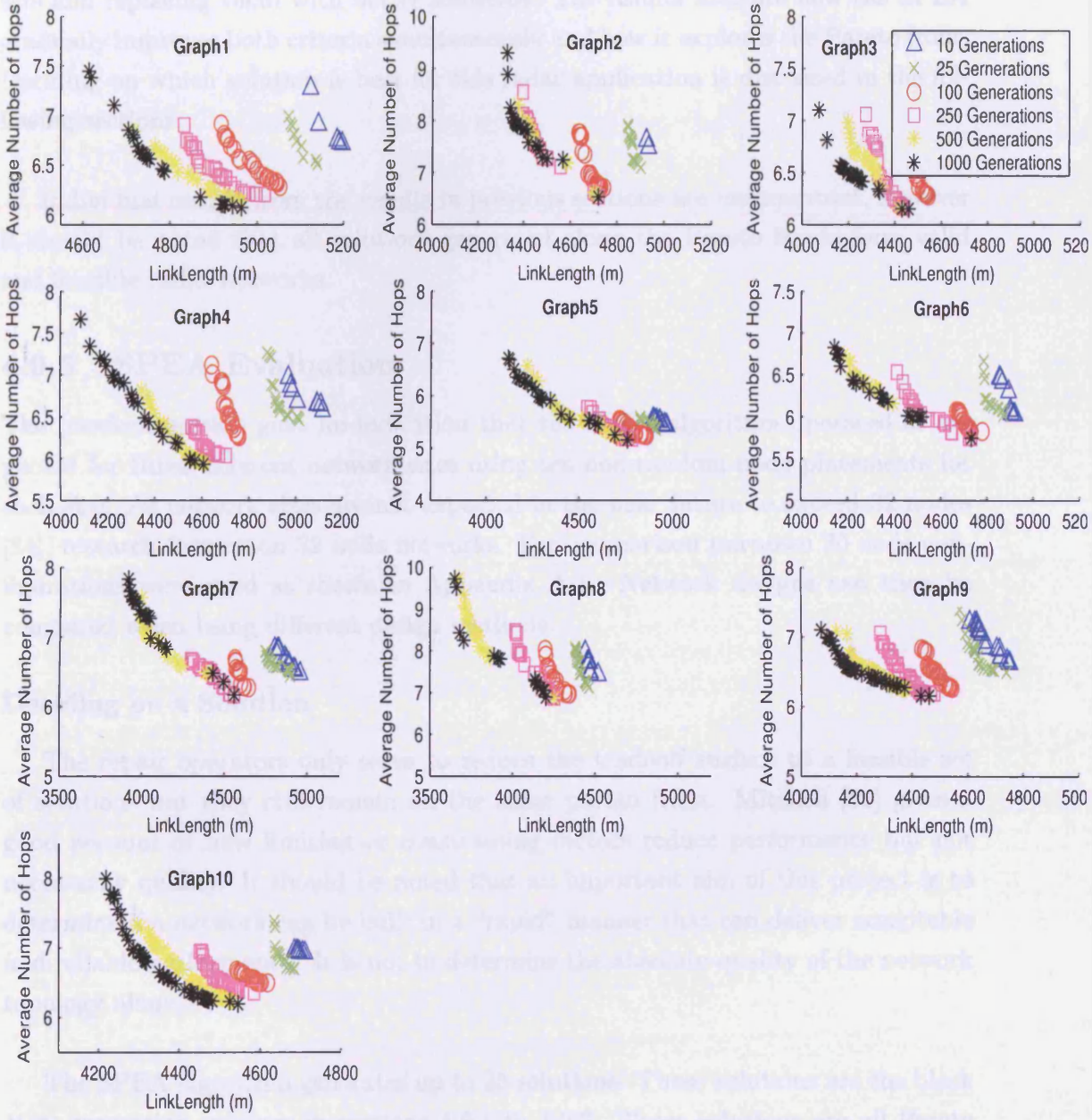



Figure 4.14: SPEA Results 128 nodes with constraints Test Graphs 1-10

### 4.9.4 Summary of SPEA results

The results shown in previous sections 4.9.1 to 4.9.3 all show the solutions gradually moving from right to left and expanding along the pareto front. The results after 1000 generations show that in all graphs the Pareto front moves significantly from the original starting positions by successively improving on previous external sets and replacing them with better solutions. The results indicate how the SPEA gradually improves both criteria simultaneously and how it explores the Pareto front. Deciding on which solution is best for this radar application is discussed in the following sections.

Individual results from the results in previous sections are unimportant, however it should be noted that all solutions generated along the Pareto fronts form valid and feasible radar networks.

### 4.9.5 SPEA Evaluation

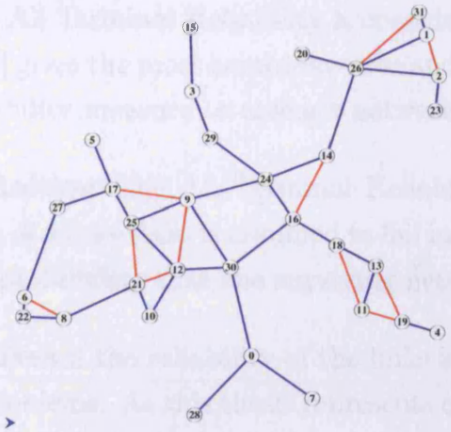
The previous section gave an indication that the SPEA algorithm operated as expected for three different network sizes using ten non-random node placements for each size. As network sizes are not expected in the near future to exceed 32 nodes [84], research focuses on 32 node networks. For comparison purposes 30 node configurations were used as shown in Appendix A.1. Network designs can then be compared when using different design methods.

#### Deciding on a Solution

The repair operators only serve to reduce the tradeoff surface to a feasible set of solutions but they still remain on the same pareto front. Mitchell [66] gives a good account of how limiting or constraining factors reduce performance but not necessarily quality. It should be noted that an important aim of this project is to determine if a network can be built in a "rapid" manner that can deliver acceptable and reliable performance. It is not to determine the absolute quality of the network topology alone.

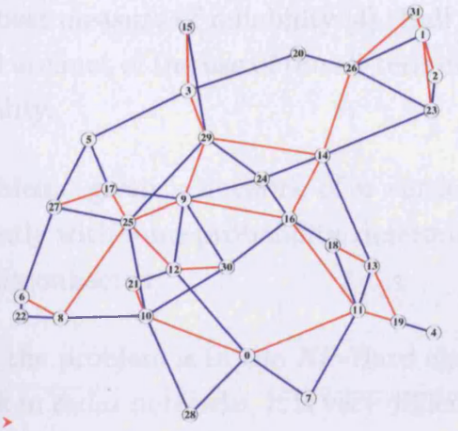
The SPEA algorithm generates up to 25 solutions. These solutions are the black 1000 generation markers in sections 4.9.1 to 4.9.3. These solutions are all Pareto optimal with respect to Average link length and hops. Determining which network is the best is a decision which can be made only by a DM (decision maker). Although the algorithm performs as required, the radar application itself renders some of the solutions invalid. Take for example, network designs of a 32 node network in figures 4.15 and 4.16. Both networks are Pareto optimal. However, whilst having fewer

high bandwidth links alone might be desirable in fixed line networks, this is not the case in these radar networks.



| Network Properties | |
|---|---|
| No. of Nodes | 32 |
| No. of Edges | 44 |
| Avg No. of Hops | 4.165 |
| Avg Link Length | 3673m |
| Avg Node Degree | 2.75 |

Figure 4.15: Pareto Example 1



| Network Properties | |
|---|---|
| No. of Nodes | 32 |
| No. of Edges | 63 |
| Avg No. of Hops | 3.03 |
| Avg Link Length | 4525m |
| Avg Node Degree | 3.94 |

Figure 4.16: Pareto Example 2

Figure 4.15 has many radar faces that have free communication timeslots which if utilised, would not affect the links already in place. Figure 4.16 is also a pareto optimal network but has all its communication faces used and therefore represents a fully optimised design for a radar network.

An option would be to add a series of modifications or additions that can be applied to a network to change its characteristics. This could be to correct the network designs to effectively fill the communication timeslots. However, this would affect the network metrics themselves and alter the network optimisation process itself. Another option would be to use some other criteria to discriminate from the already correctly designed radar networks i.e. have all/most of the communication timeslots already filled correctly by the design algorithm. The following section suggests a method for choosing a network based on criteria not directly related to the design criteria.

## 4.10 Connectivity

Radar networks require some measure that gives an indication of the reliability. The All Terminal Reliability is considered the best measure of reliability [4]. Ball et al [4] gives the most comprehensive and detailed account of the use of the all-terminal reliability measure to assess a network's reliability.

**Definition** The All Terminal Reliability Problem: given a network of $n$ vertices each of whose links is assumed to fail independently with some probability, determine the probability that the surviving network is disconnected.

Even if the reliability of the links is known, the problem is in the *NP*-Hard class of problems. As this thesis represents early work in radar networks, it is very difficult to estimate the reliability of the links and, in military networks, this is even more complicated because the links could be disrupted by external forces. Not being able to accurately estimate the reliability of the links in a radar network would reduce the usefulness of this measure.

The complexity of the All Terminal Reliability Problem also prevents it from being used with radar network design problem. Ball et al [4] does suggest surrogates for the all terminal reliability measure such as finding minimum cuts in a network. This idea has been utilised using methods that enumerate minimum cuts in networks allowing an estimate of a network's reliability to be made. David Karger's Fully Polynomial Approximation Scheme (FPAS) [56] and dynamic graph algorithms as used in sparcification techniques [69] both carry out reliability approximations using this method.

In situations where the link costs are high, such as fiber optic links, it might be appropriate to consider simpler measures such 2 edge/vertex connectivity only. i.e. if one link/node fails then there remains a path between all other nodes. 2-connectivity in optical networks is studied in [43]. There is therefore a distinction between reliable and survivable networks. Network reliability implies a quantifiable measure of absolute reliability whereas survivability implies ability to survive a specific event. Again, the main reason for the reduction of the network reliability problem to a survivability problem is due to its complexity [5].

This thesis uses the the Average All-pairs ST vertex connectivity as a surrogate for network reliability and has been defined in section 4.1. A similar measure is suggested in [70] and more recently analysed in detail in [77]. Average All-pairs ST vertex connectivity gives a good indication of the average number of vertices that have to be removed/destroyed/fail to separate any two vertices. This measure gives an indication of the connectivity of the network regardless of where the commander might be positioned. It is also calculated using simple flow methods and in polynomial time. For these reasons it is used as a reliability measure in this thesis.

### 4.10.1 Connectivity and SPEA

Previous sections analyzed the performance of the SPEA using two criteria, average number of hops and average link length. In order to analyze the performance of the SPEA using the Average All-pairs ST vertex connectivity measure, tests were done using three different optimisation methods and criteria.

1. SPEA using average no. of hops vs average link length
2. SPEA using average link length vs Connectivity
3. Single GA optimisation of Connectivity only

For each of the SPEA methods 25 candidate solutions were recorded. Only the candidate with the highest connectivity measure on the Pareto front was recorded. All time measurements were in seconds

### 4.10.2 Connectivity and SPEA Results

The following tables show how the three optimisation methods compare in terms of all-pairs ST connectivity and time taken. Each method was run for 5000 generations and the results noted. Fifty different node placements were tested. 30 were done on flat or no terrain and twenty on varying terrain from moderate or undulating to fairly severe heavy terrain [3] The best connectivity value from all twenty-five candidate solutions is shown in the tables overleaf.

[3]Obviously moderate terrain or heavy terrain are undefined and just in relation to the test area. In order to test more formally then terrain types would need to be more clearly defined.

Flat Terrain 16 Node Graphs

| Graph Num | Avg link length vs Avg hops | | Avg link length vs Connectivity | | Connectivity Only | |
|---|---|---|---|---|---|---|
| | **Connectivity** | **Time** | **Connectivity** | **Time** | **Connectivity** | **Time** |
| 1 | 3.033 | 22 | 3.033 | 978 | 3.197 | 814 |
| 2 | 3.033 | 25 | 3.058 | 926 | 3.157 | 848 |
| 3 | 3.383 | 27 | 3.425 | 1084 | 3.568 | 872 |
| 4 | 3.200 | 15 | 3.225 | 941 | 3.454 | 819 |
| 5 | 1.910 | 28 | 1.961 | 967 | 2.047 | 813 |
| 6 | 2.785 | 39 | 2.833 | 994 | 2.990 | 849 |
| 7 | 2.666 | 27 | 2.710 | 1049 | 2.853 | 843 |
| 8 | 2.410 | 24 | 2.445 | 1027 | 2.577 | 859 |
| 9 | 1.553 | 32 | 1.590 | 1014 | 1.634 | 754 |
| 10 | 2.215 | 35 | 2.250 | 909 | 2.352 | 901 |
| Minimum | 1.55 | 15.00 | 1.59 | 909.00 | 1.63 | 754.00 |
| Maximum | 3.38 | 39.00 | 3.43 | 1084.00 | 3.57 | 901.00 |
| Average | 2.62 | 27.40 | 2.65 | 988.90 | 2.78 | 837.20 |
| SD | 0.59 | 6.79 | 0.59 | 55.77 | 0.62 | 40.03 |

Table 4.1: Flat Terrain Connectivity Comparison 16 Node Graphs

Flat Terrain 32 Node Graphs

| Graph Num | Avg link length vs Avg hops | | Avg link length vs Connectivity | | Connectivity Only | |
|---|---|---|---|---|---|---|
| | **Connectivity** | **Time** | **Connectivity** | **Time** | **Connectivity** | **Time** |
| 1 | 3.603 | 191 | 3.623 | 9155 | 3.681 | 8401 |
| 2 | 2.520 | 158 | 2.542 | 9482 | 2.588 | 9181 |
| 3 | 3.468 | 169 | 3.478 | 10106 | 3.595 | 9157 |
| 4 | 2.801 | 145 | 2.863 | 9901 | 2.913 | 8151 |
| 5 | 2.907 | 157 | 2.942 | 9915 | 3.012 | 8184 |
| 6 | 3.341 | 125 | 3.364 | 9964 | 3.405 | 8841 |
| 7 | 3.212 | 159 | 3.274 | 9841 | 3.315 | 8432 |
| 8 | 3.431 | 167 | 3.455 | 9674 | 3.511 | 8123 |
| 9 | 2.839 | 135 | 2.911 | 9437 | 3.015 | 8919 |
| 10 | 3.176 | 186 | 3.248 | 9157 | 3.353 | 8848 |
| 11 | 2.951 | 175 | 3.004 | 9687 | 3.089 | 8471 |
| 12 | 3.408 | 175 | 3.455 | 9158 | 3.644 | 8393 |
| 13 | 3.066 | 169 | 3.162 | 9642 | 3.201 | 8458 |
| 14 | 3.268 | 178 | 3.331 | 10059 | 3.408 | 8194 |
| 15 | 2.942 | 169 | 3.103 | 9078 | 3.013 | 8780 |
| 16 | 2.969 | 147 | 3.050 | 9481 | 3.102 | 8709 |
| 17 | 3.268 | 165 | 3.365 | 9131 | 3.418 | 8135 |
| 18 | 2.198 | 189 | 2.265 | 9481 | 2.319 | 8486 |
| 19 | 2.782 | 139 | 2.857 | 9540 | 2.954 | 8459 |
| 20 | 3.167 | 179 | 3.197 | 9708 | 3.208 | 8131 |
| Minimum | 2.20 | 125.00 | 2.27 | 9078.00 | 2.32 | 8123.00 |
| Maximum | 3.60 | 191.00 | 3.62 | 10106.00 | 3.68 | 9181.00 |
| Average | 3.07 | 163.85 | 3.12 | 9579.85 | 3.19 | 8522.65 |
| SD | 0.34 | 18.26 | 0.33 | 326.06 | 0.35 | 338.52 |

Table 4.2: Flat Terrain Connectivity Comparison 32 Node Graphs

Moderate Terrain 32 Node Graphs

| Graph | Avg link length vs Avg hops | | Avg link length vs Connectivity | | Connectivity Only | |
|---|---|---|---|---|---|---|
| Num | **Connectivity** | **Time** | **Connectivity** | **Time** | **Connectivity** | **Time** |
| 1 | 2.515 | 125 | 2.527 | 7940 | 2.544 | 7553 |
| 2 | 2.061 | 139 | 2.120 | 8323 | 2.148 | 7855 |
| 3 | 1.713 | 198 | 1.728 | 8257 | 1.860 | 7690 |
| 4 | 1.684 | 186 | 1.711 | 8246 | 1.755 | 7892 |
| 5 | 1.788 | 189 | 1.829 | 8764 | 1.845 | 7891 |
| 6 | 1.449 | 200 | 1.472 | 8135 | 1.510 | 7314 |
| 7 | 1.726 | 197 | 1.774 | 7546 | 1.814 | 7159 |
| 8 | 1.426 | 208 | 1.484 | 8324 | 1.512 | 7429 |
| 9 | 1.104 | 195 | 1.109 | 8156 | 1.109 | 7438 |
| 10 | 2.148 | 179 | 2.193 | 8941 | 2.207 | 7919 |
| Minimum | 1.104 | 125.00 | 1.109 | 7546.00 | 1.109 | 7159.00 |
| Maximum | 2.515 | 208.00 | 2.527 | 8941.00 | 2.544 | 7919.00 |
| Average | 1.761 | 181.60 | 1.795 | 8263.05 | 1.830 | 7613.90 |
| SD | 0.403 | 27.53 | 0.406 | 389.42 | 0.406 | 274.23 |

Table 4.3: Moderate Terrain Connectivity Comparison 32 Node Graphs

Heavy Terrain 32 Node Graphs

| Graph | Avg link length vs Avg hops | | Avg link length vs Connectivity | | Connectivity Only | |
|---|---|---|---|---|---|---|
| Num | **Connectivity** | **Time** | **Connectivity** | **Time** | **Connectivity** | **Time** |
| 1 | 1.018 | 125 | 1.018 | 5940 | 1.018 | 5485 |
| 2 | 1.058 | 115 | 1.061 | 5411 | 1.061 | 5657 |
| 3 | 1.195 | 125 | 1.195 | 5748 | 1.209 | 2654 |
| 4 | 1.048 | 158 | 1.048 | 5795 | 1.052 | 5145 |
| 5 | 1.105 | 164 | 1.121 | 5816 | 1.121 | 5295 |
| 6 | 1.015 | 168 | 1.015 | 5839 | 1.019 | 5417 |
| 7 | 1.054 | 124 | 1.054 | 5719 | 1.054 | 5159 |
| 8 | 1.056 | 149 | 1.069 | 5837 | 1.069 | 5311 |
| 9 | 1.070 | 126 | 1.105 | 5948 | 1.104 | 5344 |
| 10 | 1.270 | 127 | 1.284 | 5687 | 1.291 | 5377 |
| Minimum | 1.015 | 115.000 | 1.015 | 5411.00 | 1.018 | 2654.00 |
| Maximum | 1.270 | 168.000 | 1.284 | 5948.00 | 1.291 | 5657.00 |
| Average | 1.089 | 138.100 | 1.097 | 5774.00 | 1.100 | 5084.40 |
| SD | 0.082 | 19.508 | 0.085 | 153.21 | 0.088 | 866.91 |

Table 4.4: Heavy Terrain Connectivity Comparison 32 Node Graphs

### 4.10.3 Evaluating SPEA Connectivity

Optimising for average link length vs avg hops is clearly the fastest method of designing a network. It is consistently faster by a factor of 50 or greater. In terms of connectivity, both the Avg link length vs connectivity and the Single GA connectivity optimisations outperform the Avg link length vs Avg hops optimisation. For flat terrain the simple GA connectivity outperforms the Avg link length vs Avg hops method between 1-7%. When the terrain gets heavier the number of available links decreases and therefore the number of common links increases leading to similar connectivity values.

As this is a Rapid Network Design problem, speed is the main issue. The Avg link length vs Avg hops method is significantly faster without compromising network connectivity. Using Avg link length vs Avg hops compared to Connectivity only gives a 1.01-6.11% decrease in average network connectivity. This is shown in table 4.5 which contains figures extracted from tables 4.1 - 4.4. This small variation in connectivity seems acceptable given the nature of the problem. Avg link length vs Avg hops also have much lower CPU times required to perform the optimisation.

| Optimising Criteria | Avg link length vs Avg hops Connectivity | Connectivity Only Conectivity | Percentage Difference |
|---|---|---|---|
| Flat terrain 16 Nodes (Table 4.1) | 2.62 | 2.78 | 6.11% |
| Flat terrain 32 Nodes (Table 4.2) | 3.07 | 3.19 | 3.91% |
| Moderate terrain 32 Nodes (Table 4.3) | 1.76 | 1.83 | 3.92% |
| Heavy terrain 32 Nodes (Table 4.4) | 1.09 | 1.10 | 1.01% |

Table 4.5: Percentage difference in connectivity

The difference in terms of actual network performance has not been considered. The network designs are analysed purely from a topological point of view for the different methods.

CHAPTER

# FIVE

## Radar Network Performance

Section 3 describes the role simulation plays in evaluating networks. Initial investigations are carried out on flat terrain. In order to form a representative sample of results, simulations are carried out on 30 test networks designed by the SPEA as described in section 4.9.5 page 109.

Each command position is tested for all networks resulting in a total of 960 individual simulation runs. Simulation of 30 test networks requires approximately 7 days of CPU time. Figure 5.1 below shows a sample of the 30 test networks used to evaluate the performance of the SPEA - 2 links per face. A full list of the Test Networks is available in appendix A.1. Networks chosen have the highest connectivity values in the feasible set of solutions.

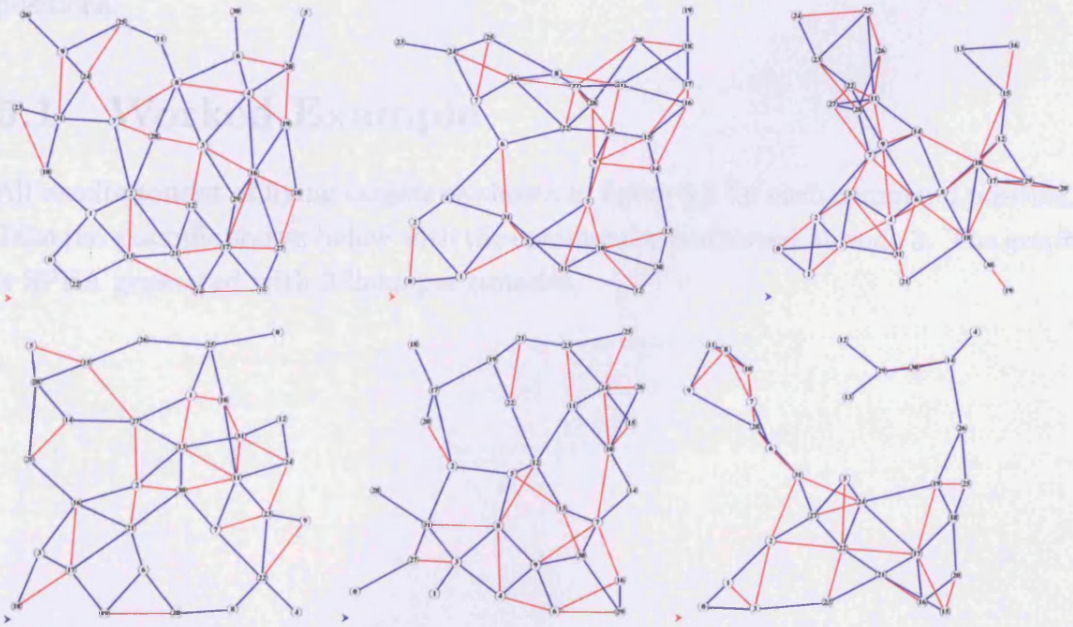


Figure 5.1: Sample of test networks

Networks are limited to 32 nodes when placed on a 28km x 28km grid. Targets are then fired across the network as shown in figure 5.2 below on a test graph. In order to keep results consistent the targets are fired simultaneously and at equal separations of 2 km.

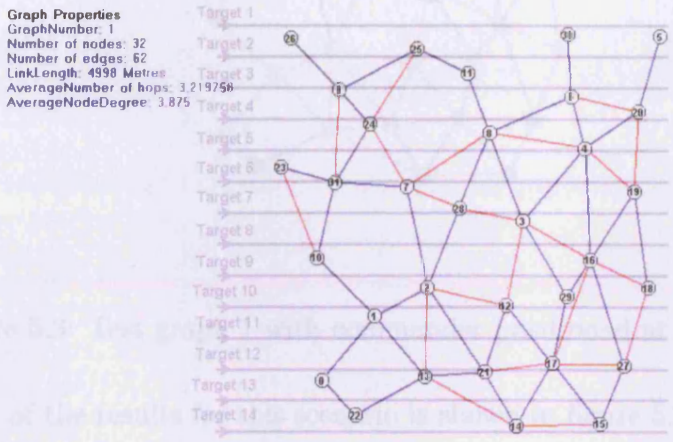



Figure 5.2: Sample network with 14 target paths

As stated in section 3.10.5, the main design objective is to create networks where average message delay from all messages generated are below 500ms. To allow full evaluation of this objective to be tested, the simulator is run using every possible command position for the 32 test networks. It is then possible to determine the quality of candidate network designs produced by the radar SPEA based on how well they perform (average message delay) for each of the 960 possible command positions.

## 5.1 Worked Example

All results consist of flying targets as shown in figure 5.2 for each command position. Take the example shown below with the commander positioned at node 3. The graph is SPEA generated with 2 links per timeslot.

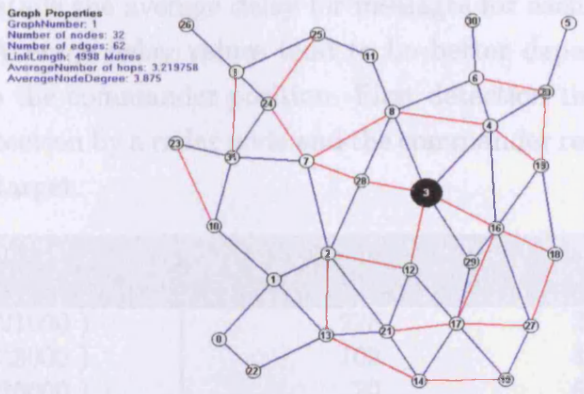



Figure 5.3: Test graph 1 with commander positioned at node 3

A summary of the results for this scenario is shown in figure 5.2 and Table 5.1. The total number of messages will remain approximately the same for all command positions. In this case the average delay value is below the target threshold of 500ms.

| | |
|---|---|
| Total Messages | 7162 |
| Average message delay | 459.27ms |
| Standard Deviation | 363.75ms |
| Min Delay | 2.00ms |
| Max Delay | 997.65ms |

Table 5.1: Performance summary

Table 5.2 details the average delay for messages for each target as it passes over the network. Average delay values tend to be better depending on the proximity of the target to the commander position. First detection times are the times taken between the detection by a radar node and the commander receiving the first message regarding this target.

| Target ID (x, y) position | First Detection Delay (ms) | No. of Messages | Avg Delay (ms) |
|---|---|---|---|
| 0(220000, 321000 ) | 228 | 392 | 431.2 |
| 1(220000, 323000 ) | 109 | 474 | 404.6 |
| 2(220000, 325000 ) | 20 | 573 | 396.7 |
| 3(220000, 327000 ) | 4 | 594 | 392.5 |
| 4(220000, 329000 ) | 2 | 593 | 414.2 |
| 5(220000, 331000 ) | 10 | 457 | 367.0 |
| 6(220000, 333000 ) | 14 | 538 | 477.7 |
| 7(220000, 335000 ) | 8 | 461 | 485.2 |
| 8(220000, 337000 ) | 26 | 501 | 478.5 |
| 9(220000, 339000 ) | 6 | 612 | 515.5 |
| 10(220000, 341000 ) | 16 | 629 | 563.2 |
| 11(220000, 343000 ) | 7 | 584 | 489.9 |
| 12(220000, 345000 ) | 10 | 456 | 489.3 |
| 13(220000, 347000 ) | 74 | 298 | 521.3 |

Table 5.2: Target statistics and delays

The following results are message delays for each node. The message delay values are consistent and the majority are below the 500ms threshold. The number of messages sent by each target is also fairly consistent as both the targets and nodes are evenly spaced.

| Node No. | Msgs Sent | Avg Delay (ms) |
|---|---|---|
| 0 | 245 | 607.16 |
| 1 | 245 | 527.81 |
| 2 | 225 | 503.95 |
| 4 | 259 | 380.00 |
| 5 | 200 | 390.94 |
| 6 | 228 | 387.29 |
| 7 | 246 | 575.09 |
| 8 | 247 | 349.91 |
| 9 | 246 | 790.88 |
| 10 | 229 | 688.47 |
| 11 | 257 | 401.12 |
| 12 | 259 | 167.55 |
| 13 | 255 | 363.79 |
| 14 | 168 | 359.46 |
| 15 | 170 | 425.53 |
| 16 | 239 | 380.89 |
| 17 | 256 | 351.96 |
| 18 | 225 | 396.56 |
| 19 | 226 | 374.19 |
| 20 | 260 | 510.84 |
| 21 | 260 | 256.28 |
| 22 | 189 | 473.76 |
| 23 | 182 | 480.81 |
| 24 | 239 | 625.21 |
| 25 | 221 | 544.11 |
| 26 | 168 | 672.63 |
| 27 | 259 | 421.62 |
| 28 | 259 | 314.66 |
| 29 | 252 | 334.80 |
| 30 | 192 | 395.02 |
| 31 | 256 | 842.91 |
| Minimum | 168 | 167.55 |
| Maximum | 260 | 842.91 |
| Average | 231 | 461.14 |
| SD | 30 | 150.99 |

Table 5.3: Node message delay Values

## 5.2 Algorithm Performance Through Simulation

The SPEA produces upto 25 different solutions and from these, one network is chosen based on methods described in section 4.10, the network with the highest connectivity values. A question arises as to whether the performance of the networks, in terms of evaluation under simulation, improves as the SPEA algorithm progresses. i.e. do the performance criteria used in evaluating networks improve as the optimisation criteria and algorithm progress/improve? Tests were carried out at different generations of the algorithm and a network design was chosen for each of the 30 test networks. Tests were carried out at the 1st, (essentially a random network with repair operators evoked) 50th, 250th, 500th and 1000th generation. The percentage of command positions was then plotted against average message delay as shown in figure 5.4.

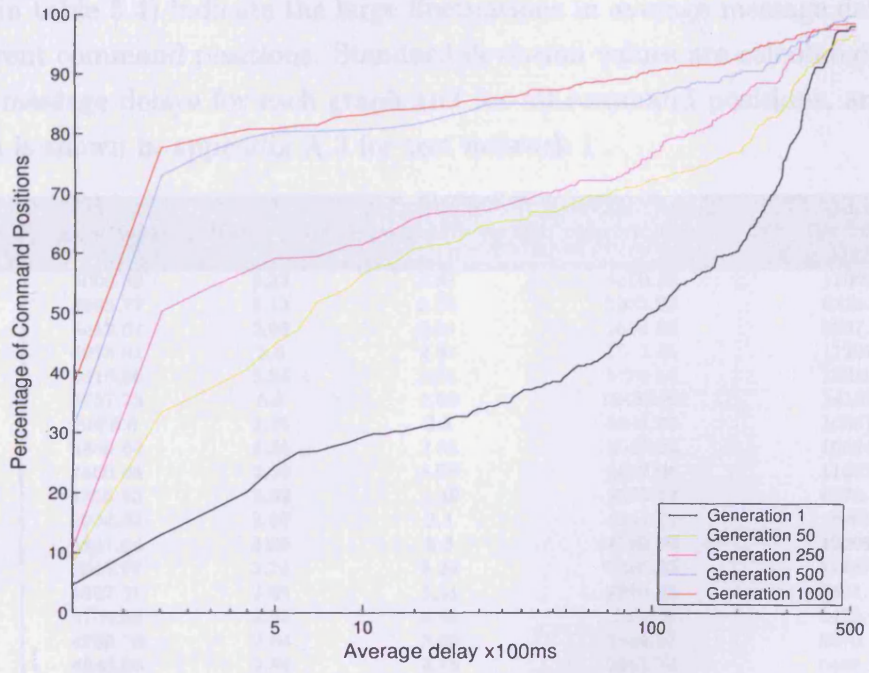



Figure 5.4: Algorithm performance

Figure 5.4 gives a clear indication that as the optimisation progresses, a larger percentage of command positions attain average message delays of less than 500ms. This does suggest that even though the SPEA algorithm does not optimise performance directly, metrics used in the design process are sympathetic to better performing networks.

*N.B. The simulations carried out in Figure 5.4 were done on different platforms due to the computationally expensive nature of the test.*

## 5.3 2 Links per Face Scenario

This section details results for SPEA designed networks. Section 5.1 illustrates a single simulation run. This section is a complete set of results for 30 test networks with 2 links per face. The complete set of network designs can be seen in appendix A.1.

### 5.3.1 Results - 2 Links per Face

Table 5.4 below shows the graph properties and average message delays of simulation runs on 30 test networks. Average message delay is calculated over all commander positions for each network generated for each starting node configuration. The average message delay values are not representative of the quality of the individual networks. The comparatively large average message delay standard deviations (last column in table 5.4) indicate the large fluctuations in average message delay values for different command positions. Standard deviation values are calculated based on average message delays for each graph and for all command positions, an example of which is shown in appendix A.3 for test network 1 .

| Graph Number | Average Link Length | Average No. of hops | All pairs ST Connectivity | Average Message Delay (ms) | Standard Deviation Avg Msg Delay |
|---|---|---|---|---|---|
| 1 | 5006.33 | 3.22 | 2.87 | 5660.33 | 11906.24 |
| 2 | 4995.77 | 3.13 | 2.88 | 3082.08 | 6126.405 |
| 3 | 4413.67 | 3.08 | 2.31 | 2664.83 | 5537.019 |
| 4 | 4975.81 | 3.3 | 2.83 | 5011.44 | 11207.64 |
| 5 | 4916.56 | 3.24 | 2.88 | 5763.58 | 12389.12 |
| 6 | 4757.73 | 3.9 | 1.66 | 10492.84 | 14151.74 |
| 7 | 5068.6 | 3.25 | 2.8 | 5047.65 | 10661.08 |
| 8 | 4889.94 | 3.35 | 2.56 | 5948.24 | 10524.22 |
| 9 | 4860.04 | 3.06 | 3.08 | 6992.08 | 11581.52 |
| 10 | 4755.85 | 2.88 | 3.58 | 2000.17 | 5173.588 |
| 11 | 4904.02 | 3.09 | 3.4 | 4241.72 | 9980.12 |
| 12 | 4847.64 | 3.09 | 3.3 | 4195.76 | 10399.15 |
| 13 | 4844.97 | 2.74 | 3.24 | 6305.85 | 11487.78 |
| 14 | 4857.31 | 2.88 | 3.51 | 2796.35 | 8694.742 |
| 15 | 4704.83 | 2.93 | 3.65 | 1590.2 | 3429.439 |
| 16 | 4798.76 | 2.66 | 3.69 | 2834.27 | 8876.109 |
| 17 | 4846.05 | 2.84 | 3.75 | 2381.76 | 6438.244 |
| 18 | 4573.92 | 2.8 | 4.09 | 3799.27 | 9632.156 |
| 19 | 4518.14 | 3 | 2.78 | 4356.71 | 12017.82 |
| 20 | 4884.76 | 2.8 | 3.39 | 2082.53 | 7005.631 |
| 21 | 4333.37 | 3.31 | 2.54 | 7463.03 | 13630.75 |
| 22 | 4691.97 | 3.15 | 2.23 | 5602.26 | 7278.614 |
| 23 | 4526.18 | 2.88 | 3.49 | 4662.56 | 10267.42 |
| 24 | 4342.61 | 2.96 | 2.59 | 6937.76 | 12019.55 |
| 25 | 4543.94 | 2.92 | 2.98 | 7364.06 | 10974.23 |
| 26 | 4566.65 | 2.99 | 2.87 | 10274.19 | 14039.45 |
| 27 | 4691.68 | 2.97 | 3.05 | 6199.08 | 13440.76 |
| 28 | 4586.58 | 3 | 2.67 | 3351.1 | 6522.425 |
| 29 | 4624.29 | 3.16 | 2.63 | 7304.1 | 12592.32 |
| 30 | 4896.41 | 3.04 | 3.13 | 2772 | 7307.16 |
| Minimum | 4333.37 | 2.66 | 1.66 | 1590.20 | |
| Maximum | 5068.60 | 3.90 | 4.09 | 10492.84 | |
| Average | 4740.81 | 3.05 | 3.01 | 4972.59 | |
| SD | 199.61 | 0.24 | 0.52 | 2301.61 | |

Table 5.4: SPEA Results - 30 networks tested on flat terrain. All networks are designed with 2 links per face

The fluctuations in message delay can be clarified using the the following test network in figure 5.5. Table 5.6 shows how the message delay values vary depending on commander position. It is important to note that the larger message delay values occur when the commander is positioned at an edge node.

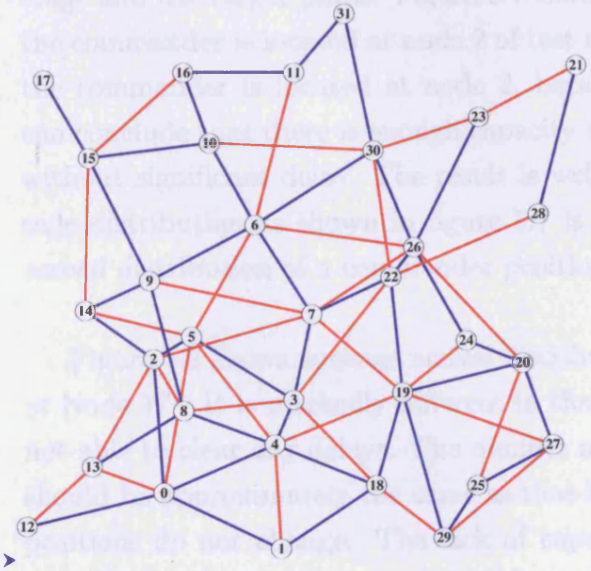



Figure 5.5: SPEA designed test network 13 with 2 links per face

| Commander Position | Avg Msg Delay (ms) |
|---|---|
| 0 | 18744.50 |
| 1 | 3309.98 |
| 2 | 357.16 |
| 3 | 291.47 |
| 4 | 302.53 |
| 5 | 337.64 |
| 6 | 291.78 |
| 7 | 273.68 |
| 8 | 349.55 |
| 9 | 368.52 |
| 10 | 414.24 |
| 11 | 12945.50 |
| 12 | 14781.80 |
| 13 | 451.14 |
| 14 | 336.57 |
| 15 | 3956.47 |
| 16 | 485.63 |
| 17 | 32679.50 |
| 18 | 350.39 |
| 19 | 283.50 |
| 20 | 408.74 |
| 21 | 32499.20 |
| 22 | 307.46 |
| 23 | 16603.10 |
| 24 | 380.94 |
| 25 | 543.48 |
| 26 | 317.35 |
| 27 | 521.15 |
| 28 | 45017.60 |
| 29 | 485.38 |
| 30 | 359.40 |
| 31 | 13031.80 |
| Minimum | 273.680 |
| Maximum | 45017.600 |
| Average | 6305.848 |
| SD | 11487.777 |

Figure 5.6: Message delay values for each commander position for test network 13

### 5.3.2 Message Arrival Analysis and Utilisation

Section 5.3.1 highlights the problem of assessing overall quality of the networks based on average delay values. The large fluctuations in average message delay is caused by lack of capacity at certain command positions, typically edge nodes as indicated by Table 5.6. These nodes tend to have adjacent edges that do not have sufficient capacity to allow messages to be transferred efficiently resulting in messages being queued for long periods of time. Higher average message delays are therefore caused by significant queueing for these under capacity links.

The message distribution is governed by the location of the nodes, their coverage and the target paths. Figure 5.7 shows the message arrival distribution when the commander is located at node 2 of test network 13 as shown in figure 5.5. When the commander is located at node 2, because there is only a 357.16ms delay, we can conclude that there is enough capacity to distribute the messages efficiently and without significant delay. The result is well below the 500ms threshold. The message distribution as shown in figure 5.7 is therefore representative of the message arrival distribution of a commander position without significant delay.

Figure 5.8 shows message arrival distribution when the commander is positioned at Node 17 . It is markedly different in that it reaches capacity very quickly and is not able to clear any delays. The number of messages and their arrival distribution should be approximately the same as that in 5.7 as both the target paths and node positions do not change. The lack of capacity limits the flow of messages in this case. Hence the maximum number of messages that can flow to the commander at position 17 is 39 messages/sec.

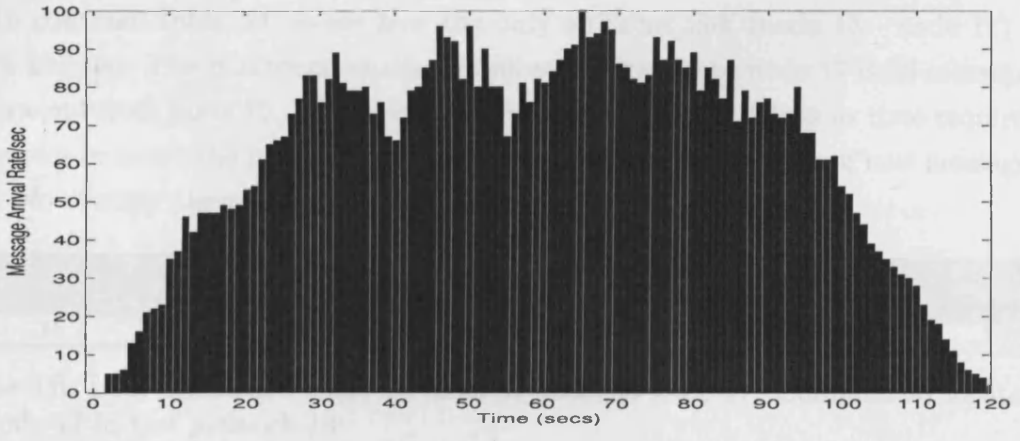



Figure 5.7: Message arrival analysis for test network 13 with commander positioned at node 2

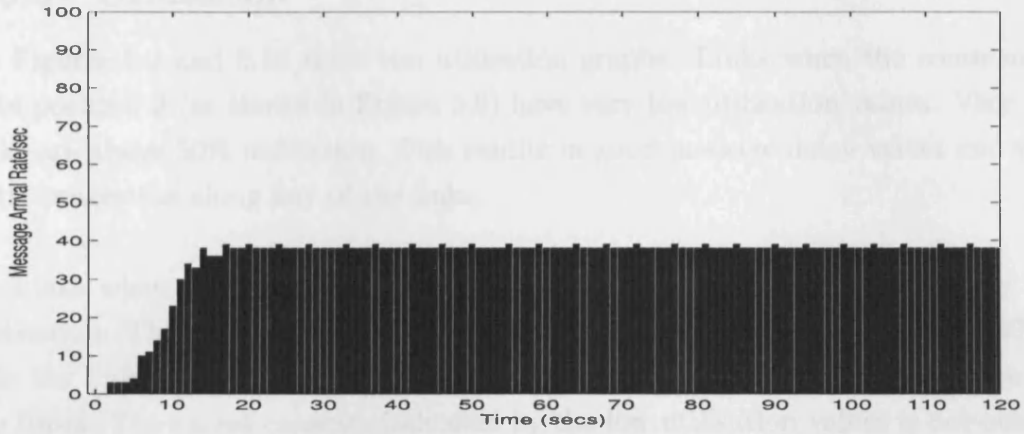



Figure 5.8: Message arrival analysis for test network 13 with commander positioned at node 17

Closer inspection shows that the low utilization values of adjacent links when the commander is at node 2 allow messages to be routed more efficiently. This is clarified in Table 5.5. None of the links adjacent to node 2 is heavily utilised.

| NodeID | Quadrant No. | Link Colour | TargetNode | Bandwidth | No. Msgs Routed | Percentage Utilisation |
|---|---|---|---|---|---|---|
| 7 | 2 | Red | 2 | 42499bps | 1054 | 69.8% |
| 5 | 2 | Blue | 2 | 596247bps | 1515 | 7.1% |
| 0 | 3 | Blue | 2 | 62137bps | 1121 | 50.7% |
| 8 | 3 | Red | 2 | 311883bps | 1762 | 15.9% |
| 13 | 0 | Red | 2 | 73919bps | 851 | 32.4% |
| 14 | 1 | Blue | 2 | 164482bps | 1162 | 19.9% |

Table 5.5: Link utilization value for links adjacent to node 2. Commander located at node 2 in test network 13

In contrast Table 5.6 shows how the only adjacent link (node 15 - node 17) is 100% utilised. The maximum number of messages routed to node 17 is 39 messages per second from node 15. Therefore there will be increasing delays as time required to service or route the messages will be higher than the arrival rate of new messages [57]. No routing algorithm will be able to compensate for this.

| NodeID | Quadrant No. | Link Colour | TargetNode | Bandwidth | No. Msgs Routed | Percentage Utilisation |
|---|---|---|---|---|---|---|
| 15 | 3 | Blue | 17 | 138615bps | 7225 | 100% |

Table 5.6: Link utilization value for links adjacent to node 17. Commander located at node 17 in test network 13

### 5.3.3 Utilisation

Figures 5.9 and 5.10 show two utilisation graphs. Links when the commander is in position 2 (as shown in Figure 5.9) have very low utilisation values. Very few links are above 50% utilisation. This results in good message delay values and very little congestion along any of the links.

Links when the commander is in position 17 (Figure 5.10) also have very low utilisation. The majority of the links have very low utilisation values (below 30%). The few links that are heavily utilised cause the dramatic increases in message delay times. The excess capacity indicated by the low utilisation values is not able to compensate for the few 100% utilised links.

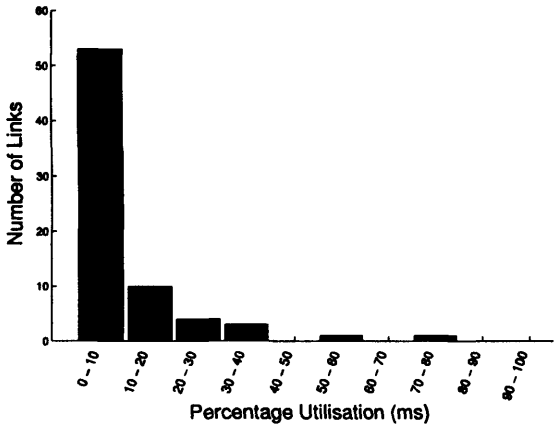



Figure 5.9: Commander positioned at node 2

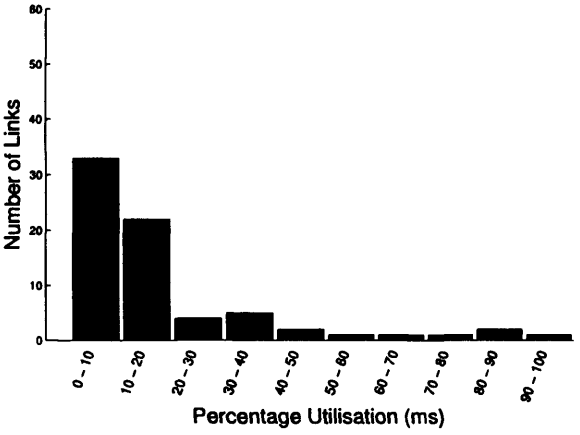



Figure 5.10: Commander positioned at node 17

These two graphs are representative of the behavior of all the networks. All graphs have low utilisation values but the concentrated traffic pattern surrounding the commander causes congestion. This issue raises the question of commander viability. Table 5.6 shows 22 out of 32 command positions achieve average message delays below 500ms. We can therefore assume that these command positions can be considered viable. This measure of assessing network performance based on the number of viable command positions is considered in the following sections.

### 5.3.4 Performance from a Military Perspective

The most important factor in determining whether the network is viable is to determine whether the commander receives messages in a timely manner. It is assumed in this thesis that this is singularly the most important criteria in measuring performance. It therefore seems appropriate to measure network performance based on the number of viable command positions in a network. We can therefore define a viable command position as a command positions that has archived an average message delay below 500ms under simulation.

Table 5.6 shows that test network 13 has 22 out of 32 command positions where there is an acceptable level of delay (below 500ms). The average delay ranges of all command positions in all test graphs is shown below.

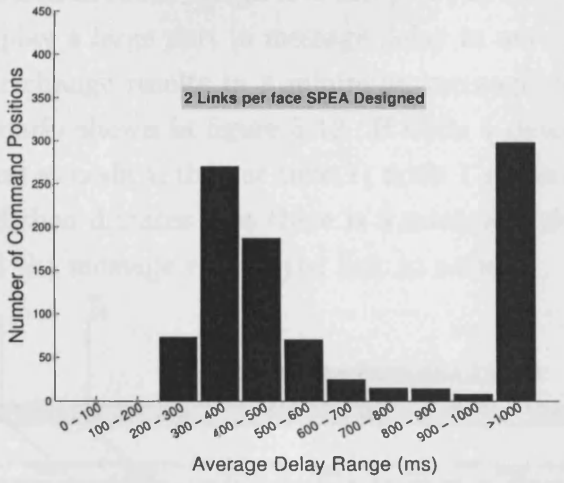



Figure 5.11: Average delay ranges for all commander positions

Figure 5.11 is considerably better at reflecting the performance of the SPEA designed networks with 2 links per face. Over 60% of command positions have acceptable performance levels as illustrated in figure 5.11. However figure 5.11 also highlights the problem area with respect to designing networks for GBAD radar systems. Commanders located at these difficult command positions would encounter large delays in the system, reducing tactical tempo and hence usefulness of the system.

### 5.3.5 Performance Analysis and Summary

The overall results as shown in Table 5.4 do not give any indication of network performance when accessing network viability. The position of the commander can have a dramatic effect on average message delay. Nodes toward the edges of the networks will tend to have fewer adjacent links and hence less capacity/routing op-

portunities. This is true of any network. Traditional network design can alleviate these problems by having much higher bandwidth infrastructure closer to the servers. For instance fiber-optic links might connect routers/switches to servers where the amount of traffic will be considerably higher than other sections of the network (further details regarding wired high bandwidth infrastructure can be found in [96]).

## 5.4 1 Link Per Face Scenario

The previous section detailed results for SPEA designed networks with 2 links per face. The primary advantage with 2 links per face is that it produces more robust networks with higher ST-connectivity values. However the 2 links per face assumption has increased network design and setup requirements such as colouring. Colour changes also play a large part in message delay in networks with 2 links per face. A single colour change results in a minimum message delay of 90ms. Take, for example, the scenario shown in figure 5.12. If node 1 detects a target and the commander is situated at node 4, then at time $T_1$ node 1 sends a message to node 2. The timeslot method then dictates that there is a minimum delay of 90ms at time $T_2$ before it can send the message via the red link to node 4.

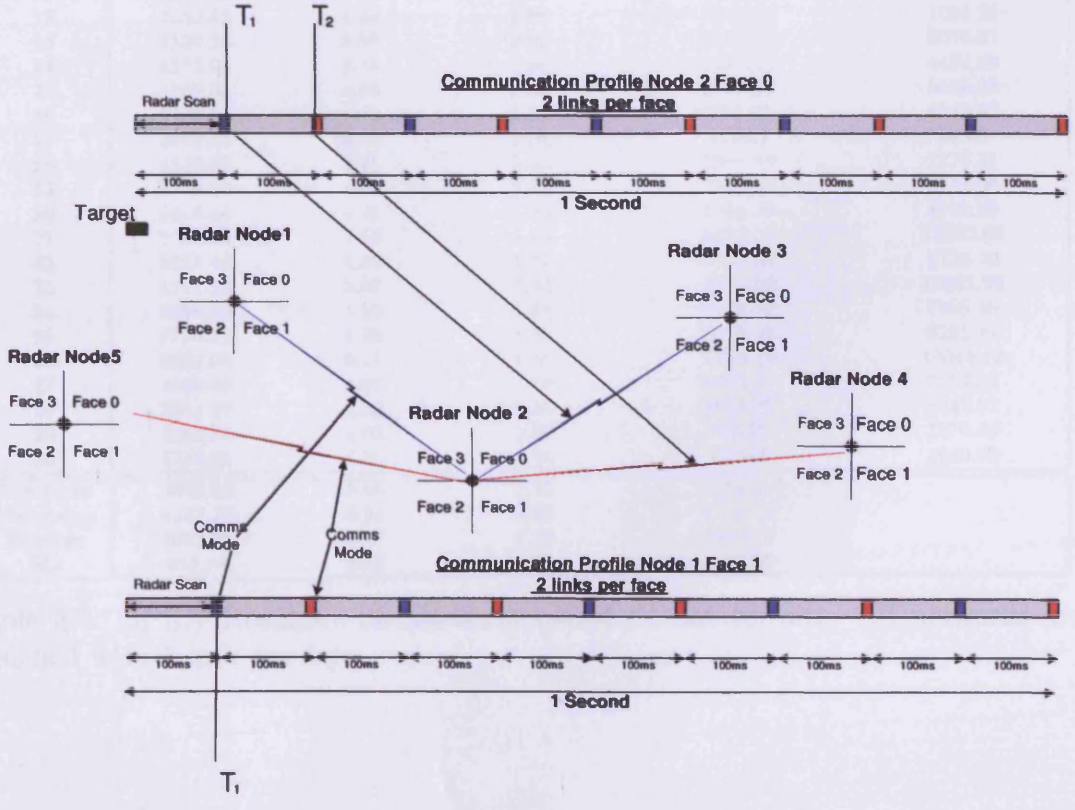



Figure 5.12: Delay caused by 2 links per face

Using 1 link per face reduces the effect of swapping timeslots but the connectivity due to the reduction in links is also reduced . However it does lead to reduced setup complexity. This section details results for SPEA designed networks with 1 link per face assumption.

### 5.4.1 Results - 1 Link per Face

As in sections 5.3.1 the SPEA is used to design 30 test networks with 1 link per face from the initial node configurations (the node positions remain the same as those used in 2 links per face). Table 5.7 shows average performance results over all commander positions. Standard deviation of average message delay is also calculated.

| Graph Number | Average Link Length | Average No. of hops | All pairs ST Connectivity | Average Message delay (ms) | Standard Deviation Avg Msg Delay |
|---|---|---|---|---|---|
| 1 | 4552.70 | 4.01 | 1.98 | 1061.97 | 3058.63 |
| 2 | 4169.11 | 4.47 | 1.70 | 4748.40 | 7784.70 |
| 3 | 3799.91 | 4.01 | 1.51 | 3276.92 | 9599.46 |
| 4 | 4633.65 | 4.16 | 1.84 | 3930.01 | 7620.11 |
| 5 | 4387.27 | 4.14 | 1.80 | 2145.67 | 6424.54 |
| 6 | 5287.73 | 3.78 | 1.10 | 4050.10 | 12078.30 |
| 7 | 4725.18 | 3.94 | 1.92 | 474.08 | 2049.61 |
| 8 | 4064.78 | 4.52 | 1.65 | 5439.06 | 8951.07 |
| 9 | 4304.33 | 3.91 | 1.72 | 2124.92 | 5899.98 |
| 10 | 3920.61 | 4.01 | 2.07 | 103.15 | 42.50 |
| 11 | 4321.43 | 4.08 | 2.09 | 857.96 | 4312.83 |
| 12 | 4230.55 | 4.34 | 1.96 | 302.78 | 1084.25 |
| 13 | 4137.24 | 3.56 | 2.02 | 1499.78 | 5076.37 |
| 14 | 4273.94 | 3.78 | 1.90 | 1437.38 | 4494.89 |
| 15 | 4059.05 | 4.05 | 2.06 | 2718.11 | 5528.33 |
| 16 | 3439.01 | 4.70 | 1.91 | 966.45 | 4215.47 |
| 17 | 3920.58 | 4.00 | 2.04 | 117.27 | 89.60 |
| 18 | 4129.50 | 3.71 | 2.23 | 1597.63 | 7376.45 |
| 19 | 3939.30 | 4.01 | 1.79 | 2326.52 | 6741.96 |
| 20 | 3914.44 | 4.21 | 1.85 | 1045.76 | 4719.65 |
| 21 | 3702.95 | 4.53 | 1.63 | 5451.52 | 12243.65 |
| 22 | 3833.46 | 4.33 | 1.51 | 3153.51 | 5723.00 |
| 23 | 4112.13 | 3.87 | 1.94 | 3772.08 | 10921.96 |
| 24 | 3696.33 | 4.50 | 1.34 | 3938.46 | 7998.86 |
| 25 | 3730.51 | 4.25 | 1.65 | 3924.40 | 9261.95 |
| 26 | 3022.04 | 6.11 | 1.30 | 7778.29 | 10044.12 |
| 27 | 4069.46 | 4.06 | 1.88 | 2659.51 | 7264.36 |
| 28 | 3911.80 | 4.20 | 1.56 | 3616.78 | 8847.77 |
| 29 | 4294.30 | 4.02 | 1.70 | 775.71 | 1570.44 |
| 30 | 4258.01 | 4.36 | 2.14 | 412.48 | 1640.60 |
| Minimum | 3022.04 | 3.56 | 1.10 | 103.15 | |
| Maximum | 5287.73 | 6.11 | 2.23 | 7778.29 | |
| Average | 4094.71 | 4.19 | 1.79 | 2523.56 | |
| SD | 413.64 | 0.45 | 0.26 | 1883.87 | |

Table 5.7: SPEA Results - 30 networks tested on flat terrain. All networks are designed with 1 link per face

The average message delays and standard deviations remain erratic as shown in table 5.7. There are, however, graphs that have acceptable average message delay values and low standard deviations ie graph 10 and 17. The average messages delays are all lower than the 2 links per face graphs in table 5.4. This indicates that there is an overall improvement in network performance.

The figure below shows SPEA designed Test network 13 with 1 link per face. All command positions are tested as in figure 5.5.

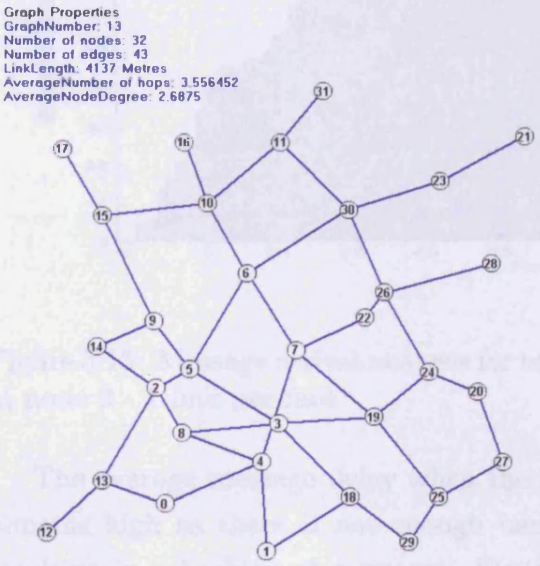



Figure 5.13: SPEA designed test network 13 with 1 link per face

| Commander Position | Avg Msg delay (ms) |
|---|---|
| 0 | 183.96 |
| 1 | 96.99 |
| 2 | 76.79 |
| 3 | 74.85 |
| 4 | 95.75 |
| 5 | 76.22 |
| 6 | 65.77 |
| 7 | 65.78 |
| 8 | 80.23 |
| 9 | 78.51 |
| 10 | 77.28 |
| 11 | 96.49 |
| 12 | 136.75 |
| 13 | 122.50 |
| 14 | 81.38 |
| 15 | 126.38 |
| 16 | 88.93 |
| 17 | 150.21 |
| 18 | 95.85 |
| 19 | 72.04 |
| 20 | 75.37 |
| 21 | 10246.90 |
| 22 | 67.82 |
| 23 | 8821.17 |
| 24 | 70.86 |
| 25 | 97.70 |
| 26 | 67.89 |
| 27 | 87.72 |
| 28 | 26228.00 |
| 29 | 90.35 |
| 30 | 86.22 |
| 31 | 110.39 |
| Minimum | 65.77 |
| Maximum | 26228.00 |
| Average | 1499.78 |
| SD | 5076.87 |

Figure 5.14: Message delay values for each commander position for test network 13

Figure 5.14 shows that although the average number of hops has increased, the average delay for each command position has decreased. Furthermore, command positions that were unusable in Figure 5.5 and Figure 5.6 have now become viable

command positions. All command positions now, apart from node 21,23 and 28, are viable command positions (a viable command position defined in section 5.3.4).

### 5.4.2 Message Arrival Analysis and Utilisation

The message arrival profile is almost identical to that in section 5.3.2 for all command positions with adequate average message delay values (below 500ms, as defined in section 3.10.5). The message arrival profile remains essentially the same shape as figure 5.7. The smoother profile is reflective of more efficient message passing and lower average delay values. The message arrival profile for test network 13 when the commander is located at position 2 is shown below in figure 5.15.

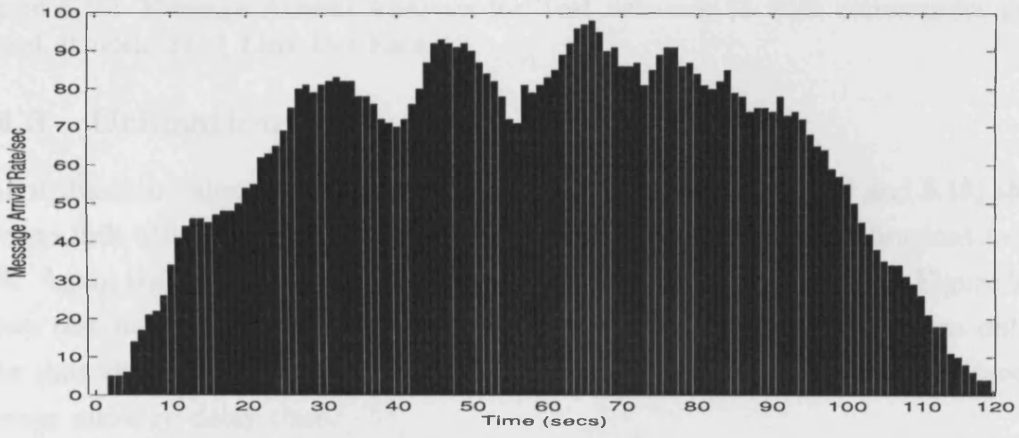



Figure 5.15: Message arrival analysis for test network 13 with commander positioned at node 2 - 1 link per face

The average message delay when the commander is located at position 21 still remains high as there is not enough bandwidth to transport messages efficiently resulting in a backlog of messages. Figure 5.16 shows the profile when delays are high. It should have a similar profile as that of figure 5.7.

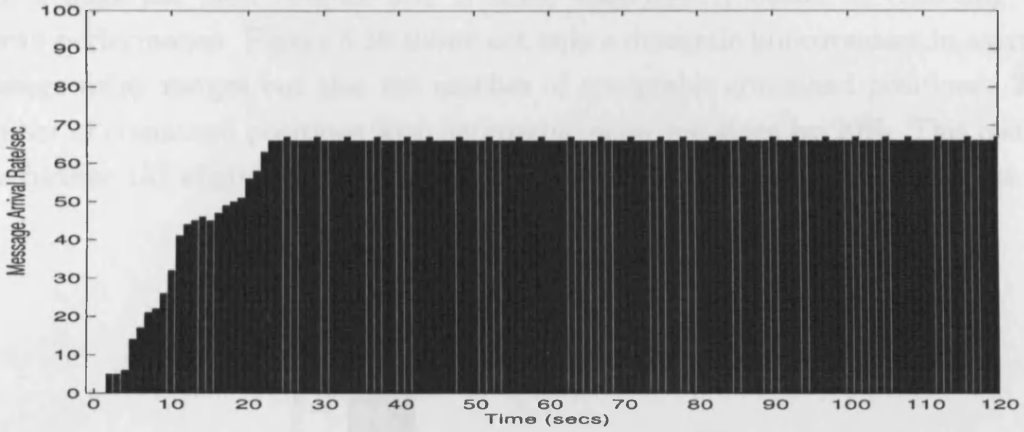



Figure 5.16: Message Arrival Analysis for Test network 13 with commander positioned at node 21- 1 Link Per Face

### 5.4.3 Utilisation

Link utilisation values remain low. Both graphs below (figures 5.17 and 5.18) show average link utilisations of test network 13. All nodes have link utilisations below 10%. Again the problem remains the few links that are heavily utilised. Figure 5.18 shows test network 13 with the commander positioned at node 21. It has only 2 links that are fully utilised (effectively a bottleneck) but this results in a 10 second average message delay time.

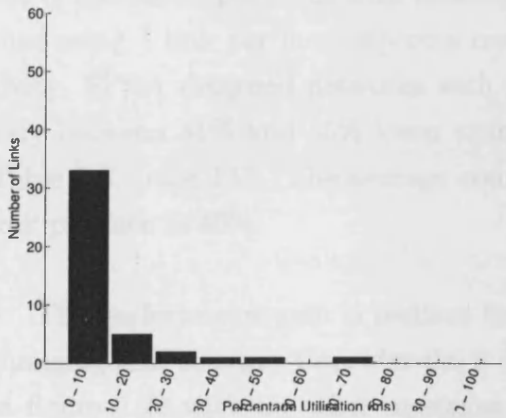



Figure 5.17: Commander positioned at node 2

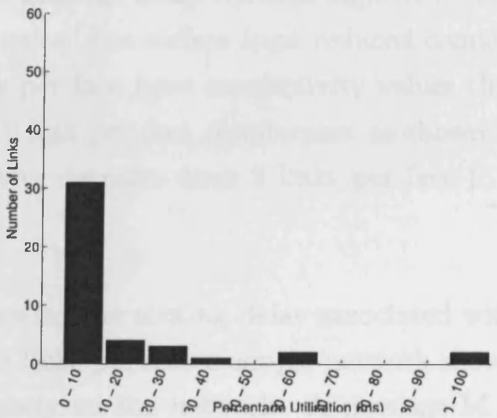



Figure 5.18: Commander positioned at node 21

### 5.4.4 Performance from a Military Perspective (1 Link per Face)

The average message delay values as shown in table 5.7 are not indicative of the improvement in performance gained in comparison with the SPEA designed networks

with 2 links per face. Figure 5.19 is again considerably better at reflecting the overall performance. Figure 5.19 shows not only a dramatic improvement in average message delay ranges but also the number of acceptable command positions. The number of command positions with acceptable delay has risen by 20%. This results in a further 137 eligible command positions taken over all 30, 32 node networks.

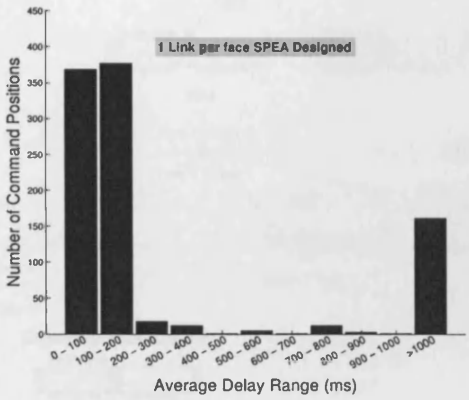



Figure 5.19: Average delay ranges for all commander positions

### 5.4.5 Performance Analysis (1 Link Per Face versus 2 Links Per Face)

Although there is a significant performance gain using 1 link per face, the number of command positions with unacceptable message delay remains high. It is clear that using 1 link per face improves results overall but suffers from reduced connectivity. SPEA designed networks with 1 link per face have connectivity values that vary between 31% and 55% lower than the 2 link per face counterpart as shown in Table 5.8, page 137. The average connectivity decrease from 2 links per face to 1 link per face is 40%.

The performance gain is realised by removing the routing delay associated with changing link colours. Consider the 6 node 2 links per face example network shown in figure 5.20 with no other messages currently on the network. A message M is produced after detecting the target shown. Message M is routed via RN3 (Radar Node 3), RN4 to arrive at RN6. The following sequence of events then occurs:

1. A target is detected at some time $T_1$ and a message M is produced.
2. At time $T_2$ the message is sent via RN1 Face1 red link to RN3.
3. The message M arrives at RN3 at time $T_{2+\epsilon}$ where $\epsilon$ is the time taken to traverse the link.
4. The next hop to RN4 requires traversing the blue link. The next blue timeslot occurs after 90ms. At time $T_3$ the message is sent via the blue timeslot on Face 0.

5. The next hop to RN6 requires traversing the red link. Again a minimum delay of 90ms occurs. At time $T_4$ the message is sent via the red timeslot on Face 1.

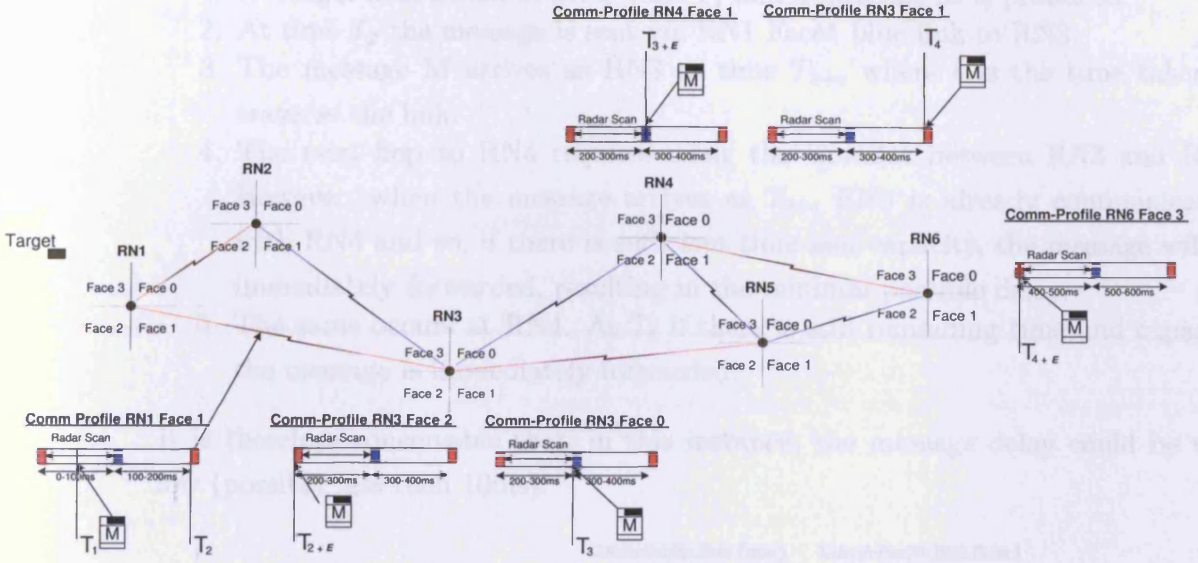



Figure 5.20: 6 Node test network, 2 links per face

The overall profile of message M is shown below in figure 5.21. Even though the network is not congested, the colour changes introduce delays because of the requirement to wait for the correct timeslot. In this case a single colour change introduces an overall minimum delay of 200ms. The problem is compounded if there is any congestion and messages are forced to wait more than one timeslot. For instance, if at time $T_3$, there were too many messages in the queue to send in a single timeslot, the message would be delayed till the following timeslot (as indicated by the red timing values in figure 5.21) which would result in a minimum delay of 290ms.

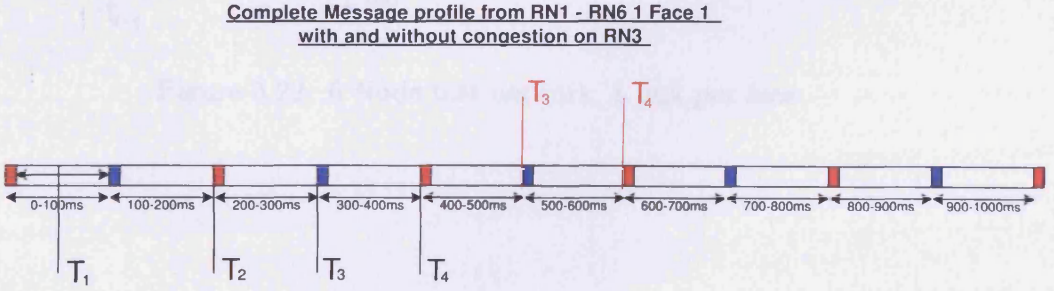



Figure 5.21: 6 Node test network - complete profile

Now consider the 6 node, 1 link per face example network shown in figure 5.22. The following sequence of events then occurs:

1. A Target is detected at some time $T_1$ and a message M is produced.
2. At time $T_2$ the message is sent via RN1 Face1 blue link to RN3.
3. The message M arrives as RN3 at time $T_{2+\epsilon}$ where $\epsilon$ is the time taken to traverse the link.
4. The next hop to RN4 requires using the timeslot between RN3 and RN4. However, when the message arrives at $T_{2+\epsilon}$ RN3 is already communicating with RN4 and so, if there is sufficient time and capacity, the message will be immediately forwarded, resulting in the minimal possible delay.
5. The same occurs at RN4. At $T_4$ if there is still remaining time and capacity, the message is immediately forwarded.

It is therefore conceivable that, in this instance, the message delay could be very low (possibly less then 10ms).

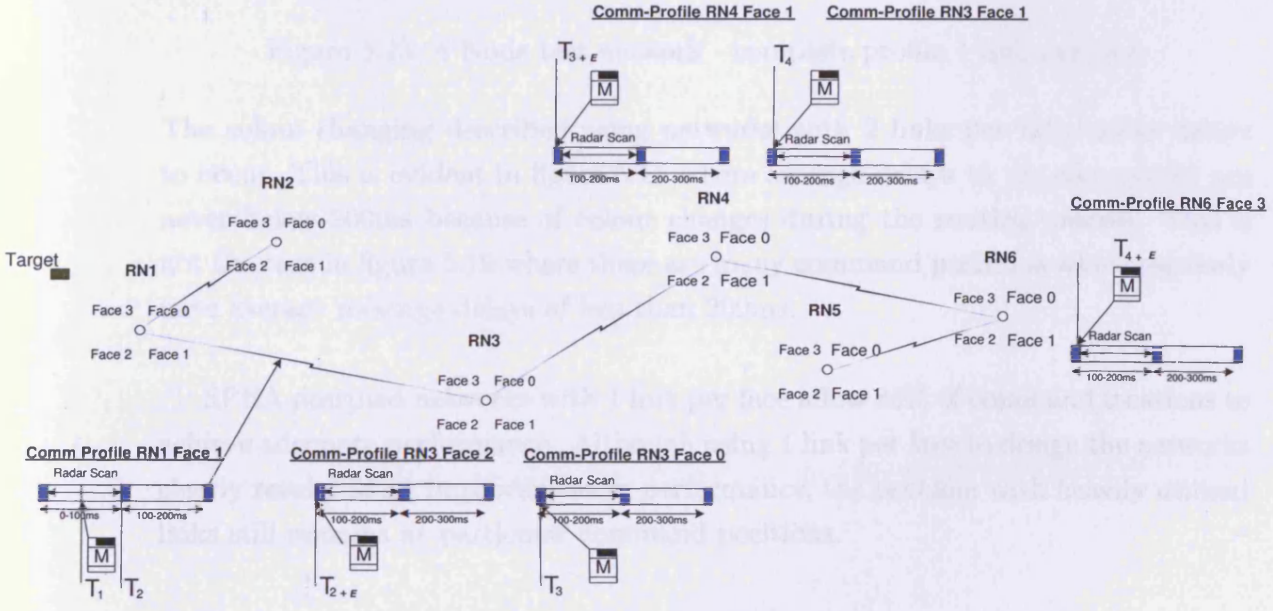



Figure 5.22: 6 Node test network, 1 link per face

The overall profile of message M is shown below in figure 5.23. Now $T_2$, $T_3$ and $T_4$ all occur in the same timeslot as they are all now routed and transmitted during this period. Even if there wasn't enough capacity or time at any particular node, because there is no colour changing, the delay would be limited to 90ms. For instance, if node RN3 didn't have enough capacity or time, then a delay of only 90ms would occur before the next available timeslot (as indicated in the red highlighted T values in figure 5.23).

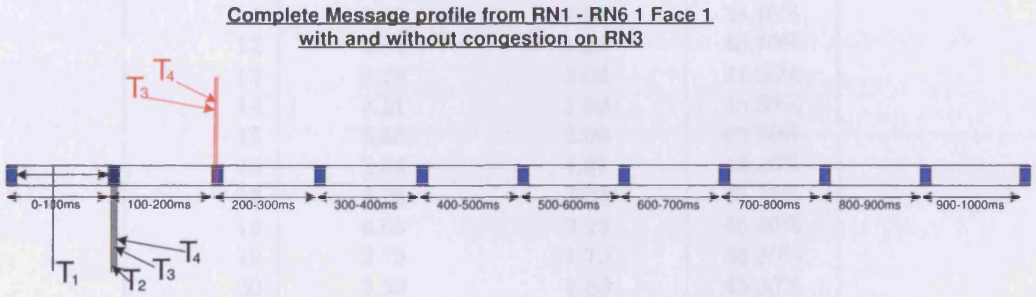



Figure 5.23: 6 Node test network - complete profile 1 link per face

The colour changing described using networks with 2 links per face causes delays to occur. This is evident in figure 5.11 where average delays to the commander are never below 200ms because of colour changes during the routing process. This is not the case in figure 5.19 where there are many command positions which regularly have average message delays of less than 200ms.

SPEA designed networks with 1 link per face allow 85% of command locations to achieve adequate performance. Although using 1 link per face to design the networks clearly results in an improvement in performance, the problem with heavily utilised links still remains at particular command positions.

| Graph No. | *Connectivity* -SPEA Designed 2 links Per Face | 1 link Per Face | Percentage Decrease |
|---|---|---|---|
| 1 | 2.87 | 1.98 | 30.90% |
| 2 | 2.88 | 1.70 | 40.90% |
| 3 | 2.32 | 1.51 | 34.60% |
| 4 | 2.83 | 1.85 | 34.80% |
| 5 | 2.88 | 1.80 | 37.50% |
| 6 | 1.66 | 1.10 | 33.70% |
| 7 | 2.80 | 1.92 | 31.70% |
| 8 | 2.56 | 1.65 | 35.60% |
| 9 | 3.08 | 1.72 | 44.10% |
| 10 | 3.59 | 2.08 | 42.10% |
| 11 | 3.40 | 2.09 | 38.70% |
| 12 | 3.30 | 1.96 | 40.70% |
| 13 | 3.24 | 2.02 | 37.50% |
| 14 | 3.51 | 1.90 | 45.80% |
| 15 | 3.65 | 2.06 | 43.70% |
| 16 | 3.69 | 1.91 | 48.40% |
| 17 | 3.75 | 2.04 | 45.70% |
| 18 | 4.09 | 2.23 | 45.50% |
| 19 | 2.78 | 1.79 | 35.80% |
| 20 | 3.39 | 1.86 | 45.30% |
| 21 | 2.54 | 1.63 | 36.00% |
| 22 | 2.23 | 1.51 | 32.40% |
| 23 | 3.49 | 1.94 | 44.60% |
| 24 | 2.59 | 1.34 | 48.40% |
| 25 | 2.98 | 1.65 | 44.80% |
| 26 | 2.87 | 1.30 | 54.50% |
| 27 | 3.05 | 1.88 | 38.30% |
| 28 | 2.67 | 1.56 | 41.70% |
| 29 | 2.63 | 1.70 | 35.20% |
| 30 | 3.13 | 2.14 | 31.60% |
| Minimum | 1.66 | 1.10 | 30.90% |
| Maximum | 4.09 | 2.23 | 54.50% |
| Average | 3.02 | 1.79 | 40.02% |
| SD | 0.52 | 0.26 | 5.95% |

Table 5.8: SPEA Designed networks - Comparison of connectivity values

### 5.4.6 SPEA Performance Summary

Radar networks can be designed with the SPEA approach but suffer from problems not directly associated with the network design process itself. Both figures 5.9 and 5.17 show the low utilisation values of the networks characterised by good network design. Also figures 5.10 and 5.18 show how individually high utilised links can cause unacceptable message delays to the commander.

SPEA designed networks with 1 link per face are the best performing but connectivity values are 30-55% lower than the 2 links per face scenarios. The choice would depend on whether the lower connectivity values were acceptable.

Figures 5.11 and 5.19 are better at illustrating overall performance but also highlight the problems associated with the networks. Obvious questions now arise:

1. Can the number of viable command positions be improved further? Are there other network design methods that can be used with this radar network design problem that can improve the number of viable command positions. Another method is considered in section 6.1.

2. Is SPEA network design using 1 or 2 links per face an appropriate method of designing radar networks? The SPEA algorithm is a global method of design but whether it is the correct method or best performing method, has not been considered. Section 6.1 considers a local approach to solving this network design problem.

3. Can command positions at difficult locations be compensated for? Modifications that can be made to improve the performance of existing network designs is detailed in section 9.

CHAPTER

# SIX

# Distributed Algorithm for Radar Topology Control

## 6.1 Decentralised and Centralised Design of Radar Networks

Using an evolutionary algorithm such as SPEA requires all information to be aggregated at a central point. This central control point uses this information in conjunction with the evolutionary algorithm to make decisions on how to best connect the network and produce a valid network design solution. Having control over the entire process allows the network topology to be analysed at the network topology level using metrics that evaluate the quality of the network. Although the information gathering and distribution have not been considered in this project, it is an important operational issue. This thesis has assumed that node positions are known to the commander and link allocations are distributed via a temporary network enabled on network startup.

Under centralised control the global central control point requires the processing power and performance to be able to carry out the network design. All SPEA algorithms consider network sizes of approximately 32 nodes allowing for very fast computation of network designs. As network sizes increase, the processing requirements, cost and time also increase. Scalability therefore becomes an issue as node sizes increase with global design methods.

From a military perspective, global design has one significant disadvantage. It provides a single point of failure to the entire network. Global design requires a central control point which is potentially the most critical node in the network. Destruction of this control node would render the network unusable as network control would be lost.

Local algorithms under decentralised control has the distinct advantage of distributing control throughout the network. This removes the need for a central control point. Network design is performed based on decisions at the node level. Each node makes decisions about links to nodes in its vicinity. The collective decision making of each node produces a network design. Whilst the same level of control cannot be maintained at a local level, there are geometrical properties that can be utilised to ensure certain network characteristics are maintained (as shown in the following sections).

## 6.2 Distributed Architecture

Local network design or distributed network design results in a loss of overall control over the entire network. Decisions have to be made at a local level in order for global networks to operate effectively [76]. Most design goals are limited to basic requirements such as [85]:

- All nodes are connected in $G$
- Minimize power or maximize energy efficiency
- Reduce interference

Topology control plays an important role in the network design. [11] gives a detailed account of the importance of energy conservation in topology control and also routing. Chang et al [11] focuses on applications that have a large number of nodes (mainly sensor networks) but low battery power and low ranges. Conversely, there are networking standards such as the IEEE 802.11 protocol, capable of being used in large networks and numbers, which rely on high bandwidth alone to maintain performance.

This project does not have the same reliance on energy conservation but does require adequate bandwidth and so requires a certain level of link length (bandwidth) control. The directional nature of the nodes used in this thesis mean that topology control more closely resembles that employed with distributed directional antennae. The tactical nature of the networks requires that some level of reliability is also maintained.

### 6.2.1 Review of Topology Control Algorithms

Ad-hoc networks or mobile networks are networks that communicate over multi-hop wireless channels. Topology control can be used to conserve energy, reduce interference and maximise capacity. There are several proposed topology control algorithms eg. [51],[86], [90], [101],[104]. These algorithms are developed with power the central issue. There are to date very few actual implemented topology control networks. Technologies such as Bluetooth and WLAN [53] rely heavily on a wired backbone in order to maintain performance. Intel[1] deployment of the IEEE 802.11 protocol highlights the importance wired backbone switches have in mobile node deployment in order to provide adequate performance. Intel also highlights the problems of peer-to-peer (multi-hop) networking where nodes themselves can become bottlenecks because of higher than expected routing requirements [18].

Ramanathan & Hain *et al* [86] uses a method of controlling topology based on locally available lists of neighbour nodes. The algorithm attempts to minimize power by minimising the node degree. It does not guarantee connectivity, allow for maximum number of links per sector, or take other geometric factors into account. Its main criteria is energy efficiency and is based on densely packed sensor networks where it is expected that each node will be in close proximity of several other nodes.

Li & Halpern[62] uses a distributed algorithm CBDTC *Cone Based Distributed Topology Control* in an attempt not only to limit the node degree but also maintain a minimal level of connectivity. The algorithm uses a progressive increase in power and properties of geometric structures such as Delaunay triangulations to achieve an acceptable level of connectivity. Its emphasis is on connectivity. Tseng [101] also provides support for maintaining the connectedness of a network and provides enhancements that allow for 2-edge and 2-vertex connected graphs.

Huang et al. [51] introduces algorithms based on methods described in [86] and [62] and applies them to directional antennae. The algorithms are further developed to reduce the number of links per sector without reducing connectivity[2]. More complex algorithms such as cluster based algorithms [90] have been recently developed to guarantee even stronger connected graphs.

[1]Intel Corp. www.intel.com

[2]Connectivity as referred here, is simply the measure that all nodes in the network are connected

### 6.2.2 DARTC Background

The DARTC algorithm was developed using knowledge gained during the development of the SPEA radar network design algorithm and combines it with traditional ad-hoc directional antenna techniques as discussed in section 6.2.1.

The DARTC algorithm attempts to maintain connectivity by maintaining links within all $\alpha$ angles, where $\alpha = 5\pi/6$. Maintaining this property guarantees connectivity. Full details and proof are given in [62]. Traditional networks use this as a lower threshold in order to minimize power. The radar networks have a fixed number of links per face and power is not a primary issue. Therefore the number of links is maximised. DARTC attempts to find at least 2 links in every $\beta$ angle where $\beta = \pi/2$ (when there are 2 links per face). Note that because $\beta$ in the DARTC algorithm is less than $\alpha$, it results in more links per face and clearly networks are more like to be connected.

As bandwidth is a function of $1/d^2$ where $d$ is the distance between nodes, short links are highly favourable. Therefore choosing short links helps preserve high bandwidth links. For this reason it seems appropriate to choose as many high bandwidth links as possible.

**Definition** A triangulation is called a Delaunay triangulation when no vertex lies within the circumscribing circle of any triangle of the triangulation.

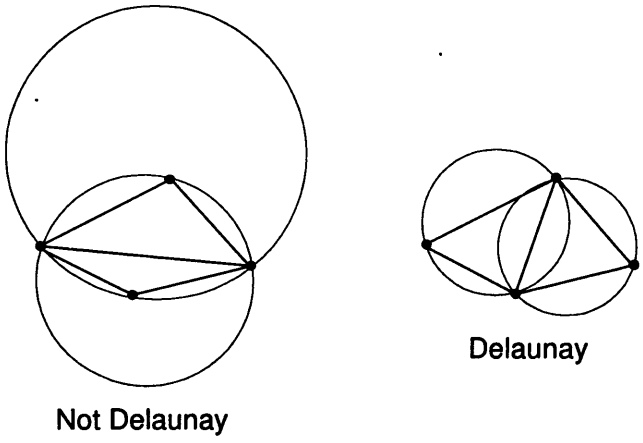



Figure 6.1: Invalid and valid Delaunay Triangulation of a 4 node network

An example of Delaunay triangulations is shown in figure 6.1. Notice how the left figure has vertices within the circumscribing circle whereas the right figure does not.

Delaunay triangulations are useful because they have several useful properties:

- For any given triangulation, Delaunay triangulations maximise the interior angle of all triangles. This will reduce interference constraints of the radars.
- Delaunay triangulations tend to produce *rounder* triangles [78]. This results in a more stable algorithm that produces *fatter* triangles as opposed to the *skinny* triangles that might be prone to interference and long links.
- The Euclidian minimum spanning tree of a network is a subgraph G of the Delaunay triangulation graph with its edges $E' \subseteq E$. The DARTC algorithm works by choosing short Delaunay edges if possible and so the probability of having this backbone of the best connecting links is higher than if Delaunay triangulations were not used. This guarantees that the highest bandwidth links are chosen.

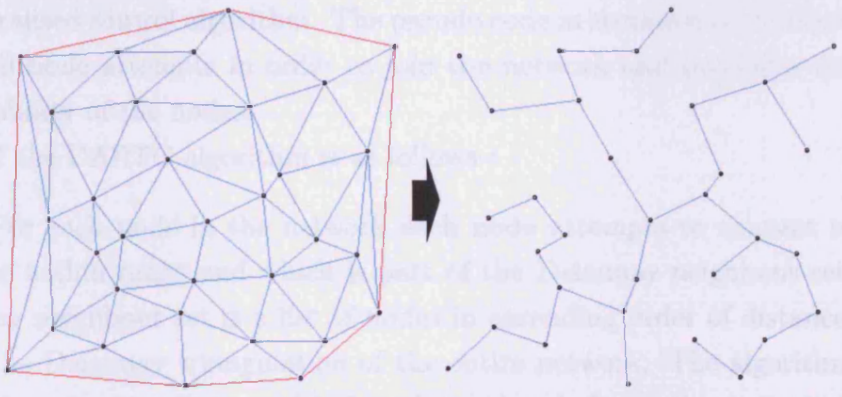


Figure 6.2: Minimum spanning subgraph of Delaunay triangulation of test network 13

The DARTC algorithm is not strictly distributed as it does require information about node locations in order to construct the Delaunay triangulation. Phase 1 of the algorithm can be relaxed to a simple nearest neighbour approach but it will not benefit from Delaunay triangulation properties.

The DARTC algorithm also requires the network and timings to be in a perfectly synchronous state. As the operation of the radars themselves is synchronous this algorithm requirement does not place additional constraints on the network.

### 6.2.3 DARTC Algorithm Description

The topology control algorithms described in Section 6.2.1 use techniques such as angle of arrival (AOA) [86] to determine node position in space. It is assumed that this project will have either embedded/attached GPS available. All links are considered symmetric to avoid confusion when using bi-directional links. It is also assumed that all nodes are aware of all node positions using a radar network discovery procedure. A radar network discovry procedure has not been considered formally in this project. It is conceivable that all nodes initially setup as many links as possible during this discovery period. Radar function could be ignored and nodes could broadcast their identity to all neighboring nodes which are in turn forwarded on to as many nodes as possible. All nodes would then be aware of node positions and possible links.

The DARTC (Distributed Algorithm for Radar Topology Control) uses ideas from the methods outlined in section 6.2.1 and from experience gained in developing and using the centralised control algorithm. The pseudo code in section 6.2.4 outlines the procedure each node attempts in order to join the network and maximise the two link per face ability of the nodes.

An overview of the DARTC algorithm is as follows :

- ***(Phase1)*** For each node in the network each node attempts to connect to another node within range and which is part of the Delaunay neighbour set. The Delaunay neighbour set is a list of nodes in ascending order of distance, built using the Delaunay triangulation of the entire network. The algorithm proceeds by first checking if the node's faces have already been allocated a link and then selecting the closest available link to add.

- ***(Phase2)*** Essentially the same as phase 1 except that the neighbour search includes all the neighbours of each node $u$. It also repeats twice in order to add a second link to each face for each node.

Delaunay neighbours are built by first building the Delaunay triangulation of the entire graph as shown in figure 6.3. Each neighbour set is then built for each node $u$ and the adjacent nodes from the Delaunay triangulation.

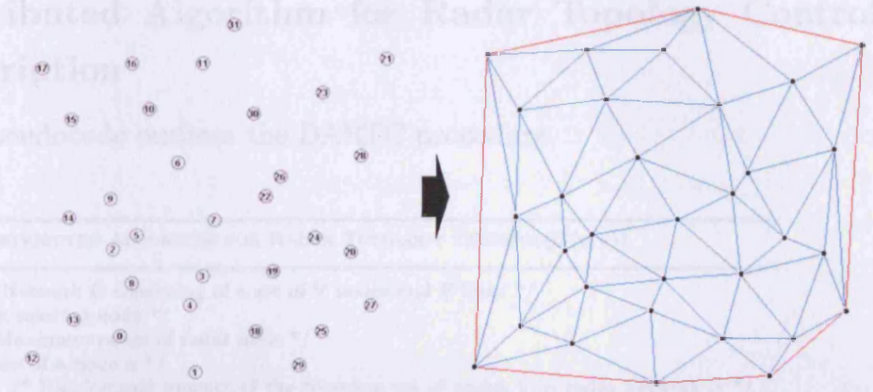


Figure 6.3: Delaunay triangulation example for Test network 13

A straightforward algorithm for calculating Delaunay triangulation can be found in [24]. It runs slightly better than linear time. However for simplicity, the LEDA triangulation method [68] was used and although it runs in linear time, it is easily integrated in the DARTC algorithm.

### 6.2.4 Distributed Algorithm for Radar Topology Control Description

The following pseudocode outlines the DARTC procedure.

**Algorithm 6.2.1:** DISTRIBUTED ALGORITHM FOR RADAR TOPOLOGY CONTROL($G(v,e)$)

```
graph G(V,E)      /* Network G consisting of a set of V nodes and E links */
node u      /* Current selected node */
MaxRange =      /* Maximum range of radar node */
face f      /* Radar face of a node u */
rV ← rand(V(G))      /* Randomised squence of the complete set of nodes V in radar network G */

/* PHASE 1 - Add single link for each face in Delaunay neighbour set */
/* For each node in random squence of nodes rV */
for each u ∈ rV
do { for each f ∈ u
     do { if Allocated(f,u) /* Check face has not already had allocated link*/
            then BREAK

          NS ← buildDelaunayNeighbourSet(u,f)/* Initalise Neighbour set */
          while t ∈ NS AND NS ≠ EMPTY
          do { if dist(u,t) ≤ MaxRange AND !LinkExists(u,t)
               then Addlink(u,t), BREAK
               else NS ← NS − t

/* PHASE 2 - Add link for each face in all neighbour set */
/* For each node in random squence of nodes rV */
for i ← 1 to 2 /* Repeat loop twice to add 2 links per face*/
do { for each u ∈ rV
     do { for each f ∈ u
          do { if Allocated(f,u,2) /* Check face has not already had 2 allocated links*/
               then BREAK

               NS ← buildNeighbourSet(u,f)/* Initalise Neighbour set */
               while t ∈ NS AND NS ≠ EMPTY
               do { if dist(u,t) ≤ MaxRange AND !LinkExists(u,t)
                    then Addlink(u,t), BREAK
                    else NS ← NS − t
```

*Functions:*
*buildDelaunayNeighbourSet*$(u,f)$ ← Builds ordered set of Delaunay neighbours of $u$ on face $f$
*buildNeighbourSet*$(u,f)$ ← Builds full ordered set of neighbours of $u$ on face $f$
*Addlink*$(u,t)$ ←Adds Link between node $u$ and node $t$
*dist*$(u,t)$ ←Returns distance between node $u$ and node $t$
*Allocated*$(f,u)$ ←Checks if face $f$ of node $u$ already has 1 allocated link
*Allocated*$(f,u,2)$ ←Checks if face $f$ of node $u$ already has 2nd allocated link

## 6.3 SPEA vs DARTC Design and Performance

The same 30 test networks used to test the SPEA were also used to test both the design and performance of the DARTC algorithm. Again the number of graphs is kept to a minimum in order to allow the simulation to run in a reasonable time[3]. Each of the 30 test networks were evaluated using DARTC with 1 and 2 links per face.

Full results for DARTC 1 and 2 links per face are available in Appendix A.4.

### 6.3.1 Network Design Comparison

Table 6.1 shows a comparison of all-pairs ST connectivity for both the DARTC and SPEA algorithms. It is clear that from a network design perspective, the global design algorithm (SPEA) is able to better control a network topology because it can utilise global information as the solutions develop. The average connectivity values of the SPEA designed networks with 2 links per face are on average 7% better than the DARTC 2 links per face. With 1 link per face SPEA designed networks are, on average, 12% better.

The SPEA all-pairs st-connectivity outperforms DARTC in all but 2 in 30 networks tested. The variance in connectivity is small ranging from a 0.0% to 20% difference in connectivity for networks built using 2 links per face. The average difference being 7%. A similar trend also appears in connectivity where networks are built using 1 link per face. When this is the case, the average difference in connectivity is 12%.

In most cases connectivity difference appears insignificant. The DARTC algorithm makes no attempt to improve the connectivity during the course of the algorithm run. The limited range of the nodes themselves is the limiting factor and accounts for the small difference in connectivity between the global and distributed algorithms.

DARTC attempts primarily to allocate the highest bandwidth links first and then adds secondary links while still being heavily biased toward short links. SPEA attempts to choose longer links to maximise hops, and as a consequence tends to have higher connectivity values[4].

[3]Testing hundreds of networks would extend simulation run-times to several months

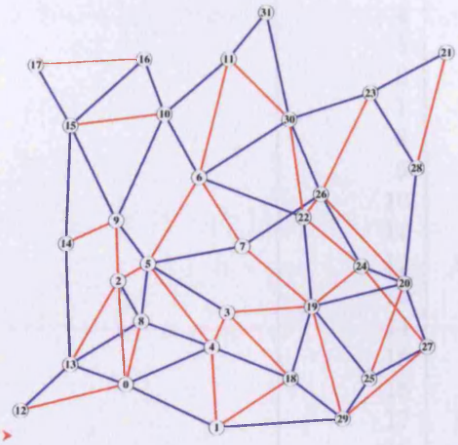
[4]The is no proven link between connectivity and node degree or hop count

| Graph No. | All Pairs ST Connectivity 2 links per face SPEA | DARTC | Percentage Difference | Graph No. | All Pairs ST Connectivity 1 link per face SPEA | DARTC | Percentage Difference |
|---|---|---|---|---|---|---|---|
| 1 | 2.87 | 2.61 | 10.06 | 1 | 1.98 | 1.72 | 15.12 |
| 2 | 2.88 | 2.87 | 0.18 | 2 | 1.70 | 1.43 | 19.27 |
| 3 | 2.32 | 2.31 | 0.05 | 3 | 1.51 | 1.50 | 1.13 |
| 4 | 2.83 | 2.54 | 11.26 | 4 | 1.85 | 1.40 | 31.84 |
| 5 | 2.88 | 2.77 | 3.92 | 5 | 1.80 | 1.67 | 7.97 |
| 6 | 1.66 | 1.61 | 2.98 | 6 | 1.10 | 1.08 | 2.15 |
| 7 | 2.80 | 2.56 | 9.45 | 7 | 1.92 | 1.93 | -0.94 |
| 8 | 2.56 | 2.34 | 9.76 | 8 | 1.65 | 1.51 | 9.37 |
| 9 | 3.08 | 2.93 | 4.88 | 9 | 1.72 | 1.61 | 7.16 |
| 10 | 3.59 | 3.45 | 4.04 | 10 | 2.08 | 2.02 | 2.59 |
| 11 | 3.40 | 2.91 | 16.98 | 11 | 2.09 | 2.02 | 3.46 |
| 12 | 3.30 | 2.99 | 10.53 | 12 | 1.96 | 1.62 | 20.92 |
| 13 | 3.24 | 3.11 | 4.22 | 13 | 2.02 | 1.86 | 8.79 |
| 14 | 3.51 | 3.42 | 2.41 | 14 | 1.90 | 1.88 | 1.28 |
| 15 | 3.65 | 3.03 | 20.72 | 15 | 2.06 | 1.39 | 47.61 |
| 16 | 3.69 | 3.61 | 2.29 | 16 | 1.91 | 1.16 | 64.35 |
| 17 | 3.75 | 3.64 | 3.13 | 17 | 2.04 | 1.69 | 20.52 |
| 18 | 4.09 | 3.39 | 20.62 | 18 | 2.23 | 2.16 | 3.06 |
| 19 | 2.78 | 2.68 | 3.84 | 19 | 1.79 | 1.64 | 9.11 |
| 20 | 3.39 | 3.36 | 1.02 | 20 | 1.86 | 1.74 | 6.33 |
| 21 | 2.54 | 2.46 | 3.61 | 21 | 1.63 | 1.57 | 3.86 |
| 22 | 2.23 | 2.22 | 0.59 | 22 | 1.51 | 1.43 | 5.65 |
| 23 | 3.49 | 2.89 | 20.95 | 23 | 1.94 | 1.92 | 0.63 |
| 24 | 2.59 | 2.53 | 2.39 | 24 | 1.34 | 1.36 | -1.90 |
| 25 | 2.98 | 2.86 | 4.45 | 25 | 1.65 | 1.52 | 8.22 |
| 26 | 2.87 | 2.76 | 3.72 | 26 | 1.30 | 1.11 | 17.30 |
| 27 | 3.05 | 2.63 | 16.05 | 27 | 1.88 | 1.62 | 16.06 |
| 28 | 2.67 | 2.46 | 8.78 | 28 | 1.56 | 1.43 | 9.18 |
| 29 | 2.63 | 2.55 | 2.97 | 29 | 1.70 | 1.68 | 1.23 |
| 30 | 3.13 | 2.79 | 12.05 | 30 | 2.14 | 1.72 | 24.85 |
| Minimum | 1.66 | 1.61 | 0.05 | | 1.10 | 1.08 | -1.90 |
| Maximum | 4.09 | 3.64 | 20.95 | | 2.23 | 2.16 | 64.35 |
| Average | 3.02 | 2.81 | 7.26 | | 1.79 | 1.61 | 12.21 |
| SD | 0.52 | 0.45 | 6.39 | | 0.26 | 0.27 | 14.66 |

Table 6.1: Comparison of SPEA vs DARTC Connectivity Results

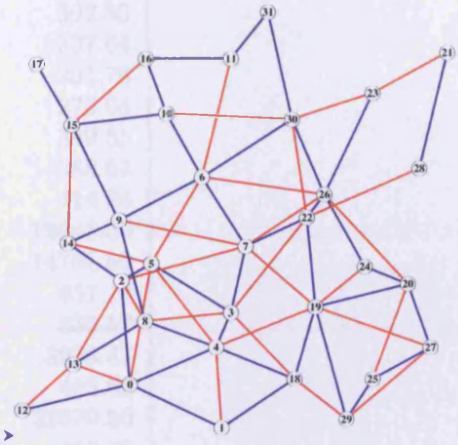
### 6.3.2 Network Performance Comparison 2 Links per Face

The figures 10.7 and10.8 below show two distinct network designs from SPEA and DARTC.



| Network Properties | |
|---|---|
| No. of Nodes | 32 |
| No. of Edges | 71 |
| No. of Hops | 2.951 |
| Avg Link Length | 4669m |
| Avg Node Degree | 4.44 |
| ST Connectivity | 3.11 |

Figure 6.4: DARTC 2LPF design



| Network Properties | |
|---|---|
| No. of Nodes | 32 |
| No. of Edges | 72 |
| No. of Hops | 2.737 |
| Avg Link Length | 4844m |
| Avg Node Degree | 4.5 |
| ST Connectivity | 3.23 |

Figure 6.5: SPEA 2LPF design

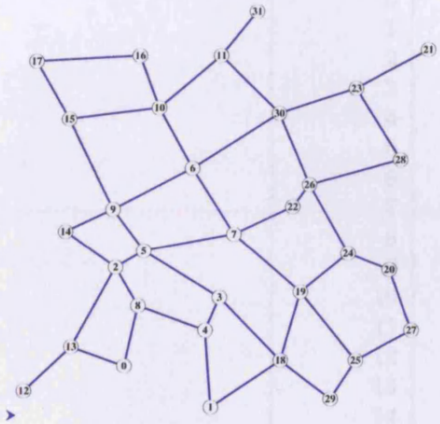
The DARTC designed network has improved message delays in 24 out of the 32 command positions. Table 6.2 150 illustrates this. Minimum, maximum and average delay values for the DARTC 2LPF network designs are all comparatively better than the SPEA 2LPF designed networks.

| Comm ID | Avg Msg delay(ms) DARTC | SPEA |
|---|---|---|
| 0 | 9626.00 | 18744.50 |
| 1 | 490.39 | 3309.98 |
| 2 | 306.32 | 357.16 |
| 3 | 360.18 | 291.47 |
| 4 | 298.26 | 302.53 |
| 5 | 269.87 | 337.64 |
| 6 | 303.82 | 291.78 |
| 7 | 349.28 | 273.68 |
| 8 | 310.36 | 349.55 |
| 9 | 301.03 | 368.52 |
| 10 | 368.66 | 414.24 |
| 11 | 568.58 | 12945.50 |
| 12 | 14298.30 | 14781.80 |
| 13 | 1483.45 | 451.14 |
| 14 | 619.07 | 336.57 |
| 15 | 616.29 | 3956.47 |
| 16 | 524.39 | 485.63 |
| 17 | 17129.00 | 32679.50 |
| 18 | 343.30 | 350.39 |
| 19 | 260.37 | 283.50 |
| 20 | 361.33 | 408.74 |
| 21 | 32230.90 | 32499.20 |
| 22 | 340.95 | 307.46 |
| 23 | 14123.20 | 16603.10 |
| 24 | 315.81 | 380.94 |
| 25 | 411.79 | 543.48 |
| 26 | 331.75 | 317.35 |
| 27 | 449.82 | 521.15 |
| 28 | 20545.40 | 45017.60 |
| 29 | 399.27 | 485.38 |
| 30 | 318.50 | 359.40 |
| 31 | 5166.97 | 13031.80 |
| Minimum | 260.37 | 273.68 |
| Maximum | 32230.90 | 45017.60 |
| Average | 3869.46 | 6305.85 |
| SD | 7629.49 | 11487.78 |

Table 6.2: Delay values SPEA 2LPF vs DARTC 2LPF *Test network 13*

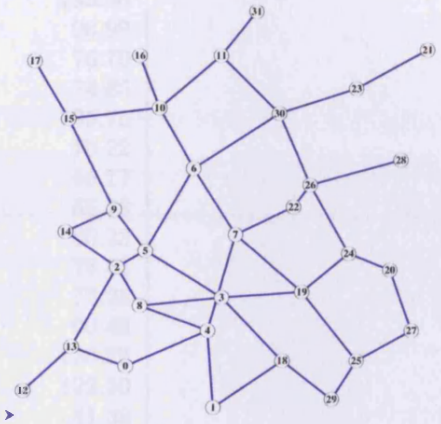
### 6.3.3 Network Performance Comparison 1 Link per Face

Figures 6.6 and 6.7 below show two distinct network designs from SPEA and DARTC designed with 1 link per face. The DARTC algorithm is modified to design 1 link per face simply by running Phase 2 once.



| Network Properties | |
|---|---|
| No. of Nodes | 32 |
| No. of Edges | 45 |
| No. of Hops | 3.78 |
| Avg Link Length | 4067m |
| Avg Node Degree | 2.85 |
| ST Connectivity | 1.86 |

Figure 6.6: DARTC 1LPF design



| Network Properties | |
|---|---|
| No. of Nodes | 32 |
| No. of Edges | 43 |
| No. of Hops | 3.57 |
| Avg Link Length | 4137m |
| Avg Node Degree | 2.68 |
| ST Connectivity | 2.02 |

Figure 6.7: SPEA 1LPF design

The results in terms of message delay times do not vary considerably between DARTC and SPEA. The comparison of commander position average message delay times as shown in Table 6.3, illustrates this. Node 21 still remains unusable.

| Comm ID | Avg Msg delay(ms) DARTC | SPEA |
|---|---|---|
| 0 | 94.06 | 183.96 |
| 1 | 104.95 | 96.99 |
| 2 | 73.68 | 76.79 |
| 3 | 86.44 | 74.85 |
| 4 | 84.49 | 95.75 |
| 5 | 72.04 | 76.22 |
| 6 | 69.00 | 65.77 |
| 7 | 65.51 | 65.78 |
| 8 | 78.46 | 80.23 |
| 9 | 71.38 | 78.51 |
| 10 | 81.63 | 77.28 |
| 11 | 97.9737 | 96.49 |
| 12 | 136.756 | 136.75 |
| 13 | 116.574 | 122.50 |
| 14 | 79.4251 | 81.38 |
| 15 | 121.882 | 126.38 |
| 16 | 97.2768 | 88.93 |
| 17 | 128.465 | 150.21 |
| 18 | 81.8632 | 95.85 |
| 19 | 74.1209 | 72.04 |
| 20 | 78.7427 | 75.37 |
| 21 | 10214.5 | 10246.90 |
| 22 | 65.3021 | 67.82 |
| 23 | 124.508 | 8821.17 |
| 24 | 73.0729 | 70.86 |
| 25 | 105.54 | 97.7 |
| 26 | 66.5299 | 67.89 |
| 27 | 92.9797 | 87.72 |
| 28 | 138.889 | 26228 |
| 29 | 89.6115 | 90.35 |
| 30 | 87.5034 | 86.22 |
| 31 | 111.37 | 110.39 |
| Minimum | 65.30 | 65.77 |
| Maximum | 10214.50 | 26228.00 |
| Average | 408.27 | 1499.78 |
| SD | 1789.56 | 5076.87 |

Table 6.3: Delay values SPEA 1LPF vs DARTC 1LPF *Test network 13*

### 6.3.4 DARTC Performance Analysis

The average message delays as shown in A.4 are again not representative of the performance of the networks. The number of viable command positions is significantly higher when networks are designed using DARTC with 1 link per face. In particular 59% of command positions designed with DARTC 2 links per face were viable. 88% of command positions designed with DARTC 1 link per face were viable. This compares to 81% and 54% of viable command positions with SPEA 1 link per face and 2 links per face respectively.

Using DARTC (1 link per face) not only has a larger number of viable command positions but the average delay of viable command positions is less than 200ms. However there is still a problem with certain command positions. This can be seen in figure 6.9 where a large proportion of command positions is in the lower delay ranges. This is in contrast to figure 6.8 which has no links in the 0-100 or the 100-200ms ranges.

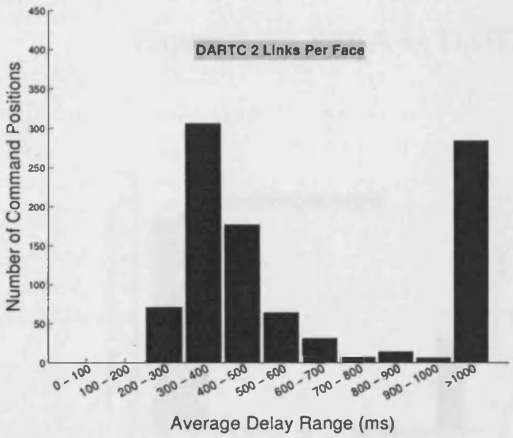



Figure 6.8: DARTC 2 Links per Face

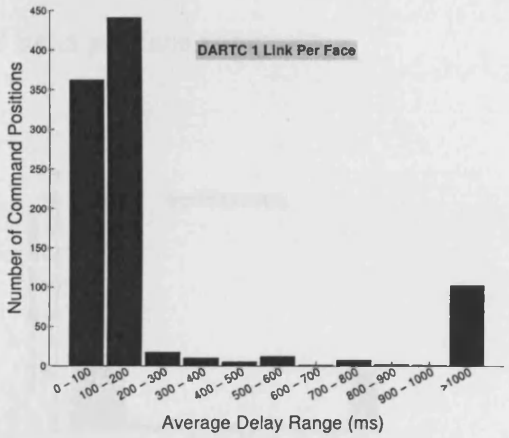



Figure 6.9: DARTC 1 Link per Face

## 6.4 DARTC Summary

On inspection DARTC outperforms SPEA using both 1 and 2 links per face. Figures 6.10 and 6.11 below show a comparison of SPEA vs DARTC using 1 and 2 links per face. Both have similar characteristics but DARTC has a larger number of viable command positions. It is also clear that using 1 link per face with either method offers significantly improved average message delays.

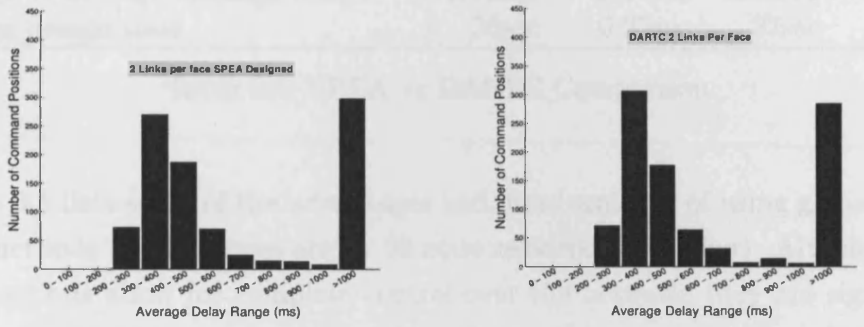



Figure 6.10: SPEA vs DARTC 2 links per face comparison

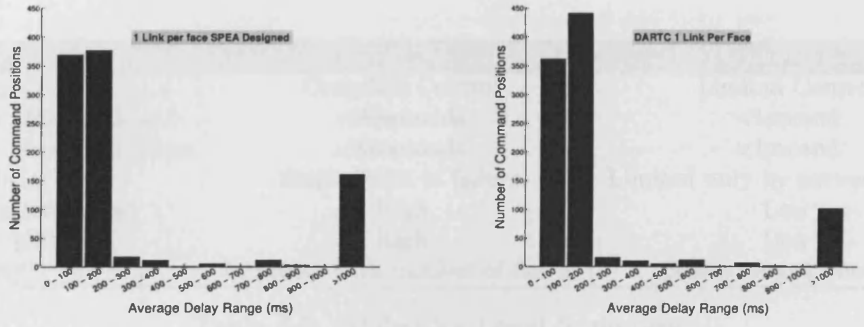



Figure 6.11: SPEA vs DARTC 1 link per face comparison

A summary of the two design methods is shown in Table 6.4. This is essentially a local vs global design comparison. There is an obvious advantage in run times with the DARTC algorithm running over 140 times faster [5]. When using either 1 or 2 links per face, the SPEA does have improved connectivity (see Table 6.1). However the DARTC performance has a 2% improvement over the SPEA when using 2 links per face and a 7% improvement when using 1 link per face. The performance increase can be attributed to selecting the higher bandwidth links as a priority in the DARTC algorithm. Average message delay values are also better using the DARTC designs.

[5]The actual runtime would be faster as the algorithm itself would be distributed to all nodes. The running order would be closer to O(1) rather than its current O(n)

Therefore from Table 6.4 DARTC can be considered an appropriate method for designing fully connected radar networks.

| | 2 Links per Face | | 1 Link per Face | |
|---|---|---|---|---|
| | SPEA | DARTC | SPEA | DARTC |
| Number of command positions tested | 960 | 960 | 960 | 960 |
| Number of viable command positions | 662 | 676 | 799 | 858 |
| Percentage viable command positions | 68.96% | 70.42% | 83.23% | 89.38% |
| Average all pairs ST connectivity | 3.01 | 2.81 | 1.79 | 1.61 |
| Average of average message delays | 4972.59 | 4017.62 | 2523.56 | 1558.92 |
| Average design time | 35sec | 0.25sec | 37sec | 0.21sec |

Table 6.4: SPEA vs DARTC Comparison

Table 6.5 lists some of the advantages and disadvantages of using global vs local control methods (design times are for 32 node networks or smaller). Although global design methods allow for complete control over the network, they are significantly slower and with higher computational overheads than the equivalent local design methods. The speed of the local design methods also scales well. Whilst this thesis has kept reliability an important issue with respect to network design, global design by definition requires a central control point, and in so doing provides a single point of failure.

| | Global Design | Local Design |
|---|---|---|
| Topology Control | Complete Control | Limited Control |
| Network Design Time | <60seconds | <1second |
| Network Re-design Time | <60seconds | <1second |
| Vulnerability | Single Point of failure | Limited only by network design |
| Computational Cost | High | Low |
| Setup Overhead | High | Low |
| Scalability | Increases with number of nodes | Scales almost linearly |

Table 6.5: Global Vs Local Summary

CHAPTER

# SEVEN

## Increasing Connectivity Through Backup Routes

Previous chapters have highlighted that optimisation algorithms using 1 link per face radar is the most effective method of constructing radar networks, the obvious disadvantage being the resulting loss in st-connectivity. This section describes a method using backup or fail-over links for improving the all-pairs st connectivity values for all radar networks with 1 link per face .

Fail-over switching has existed in *Asynchronous Transfer Mode* (ATM) [1] networks and internetworks for many years [96]. Using other routes or establishing new routes is common practice in the commercial sector today. In fixed network architectures, backup routes/hardware are used to provide failsafe and sometimes low cost options.

## 7.1 Wired Backup Example

The following example figure 7.1 page 157 is an example of how backup routes and backup hardware are commonly used in the commercial sector. Figure 7.1 shows two networks connected with current CISCO [2] internetworking infrastructure common to medium-large size enterprises. Network A is connected to network B via a CISCO 7000 [3] series router and high bandwidth primary-rate ISDN line i.e. upto 2048kbps (which could be any high bandwidth line such as ADSL [4] line). It also has a dial-on-demand CISCO 801 router that is connected to a low bandwidth ISDN 2 line ie 128kbps. The primary route in figure 7.1 represents a permanent connection to network B. The primary route is also considerably more expensive

[1] Asynchronous Transfer Mode is a network technology based on packets or cells

[2] http://www.cisco.com

[3] See CISCO web site for full details if required http://www.cisco.com/en/US/products/index.html

[4] asymmetric digital subscriber line is a technology that allows relatively large amounts of data to be sent over existing copper telephone lines

than the backup route. The actual bandwidths are insignificant as they are for illustrative purposes only.

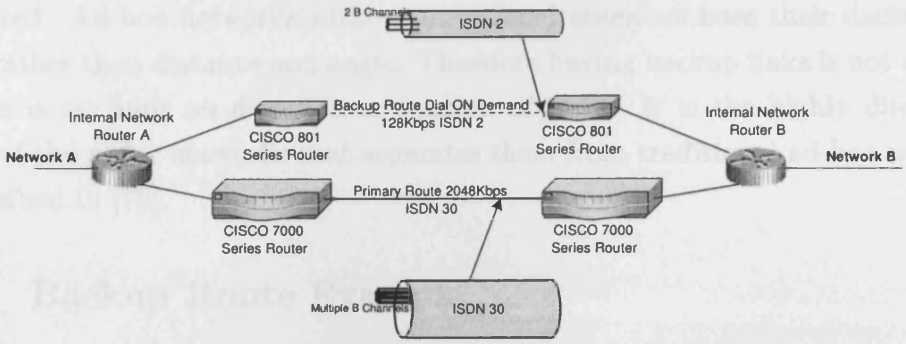



Figure 7.1: Fault Tolerance using ISDN

By configuring the network in this manner, there is a fail-over route in the event of failure in either the primary route routers or the telecom lines. For example, the Network A CISCO 7000 router fails, causing the internal network router A to send packets along to the CISCO 801 router located on the network A. A dial-on-demand link is then initiated between both CISCO 801 routers on networks A and B. All packets routed from network A to network B are then routed through the backup route until the primary route can be restored. Once the primary route has been re-established the backup link disconnects.

There are two important aspects to the example in Figure 7.1. Firstly, and most importantly, the system remains operational in the event of a significant failure. Secondly, cost is kept to a minimum in order to provide fail-over, however performance and available bandwidth is reduced.

### 7.1.1 Routing Issues in Fail-over Performance

When primary routes fail, a major factor in network performance during the transient state of the network is the routing [103]. This is an extensive subject and congestion control during failure is not considered in this thesis. Ad-hoc networks have more recently become the focus of research into congestion control during failure[12]. In this thesis, packets destined for a failed link/node will be simply moved to the next best route to the destination if available.

## 7.2 Backup Routes in Radar Networks

Applying backup route and links in traditional wireless networks is not usually considered. Ad-hoc networks without directional antennae base their decisions on power rather than distance and angle. Therefore having backup links is not an issue as there is no limit on direction or number of links. It is the highly directional nature of the radar networks that separates them from traditional ad-hoc networks as described in [76].

### 7.2.1 Backup Route Example

Consider the test network in figure 7.2 and further consider the subgraph consisting of nodes 6, 7, 22, 26. The test network is for illustrative purposes only.

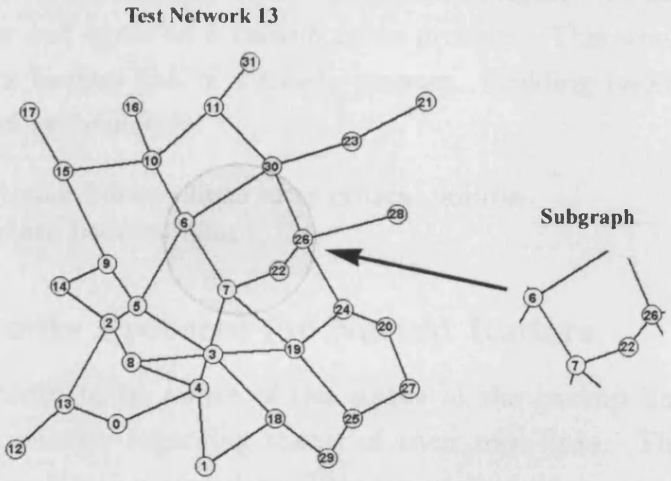



Figure 7.2: Subgraph identification for use with backup route examples

Figure 7.3 shows the subgraph with possible backup routes. If in this case, either node 7 or 22 fail, then a backup link could be setup and used as a primary route. In figure 7.3, node 7 fails and a backup route between nodes 6 and 22 is used as a primary route.


Figure 7.3: Failure in node 7 causes a backup link to become a primary link

### 7.2.2 Backup Route Setup in Radar Networks

In order for fail-over to occur for a radar network as in figure 7.3, both nodes would have to be aware and agree on a backup route protocol. This would enable both nodes to initiate a backup link in a timely manner. Building backup routes therefore centers around two concepts;

1. How to initialise backup links after critical failures.
2. Where to place backup links.

### 7.2.3 Backup Route Protocol for Netted Radars

In order for both nodes to be aware of the status of the backup links, both must communicate information regarding status of their own links. This would require a separate communication timeslot. This timeslot could be every 500ms to guarantee that backup links are brought up within the 500ms threshold. The backup communication timeslots could be made more frequent. However this would be at the expense of increased overheads. Figure 7.4 represents the communication profiles of nodes 6 and 22 in Figure 7.3.

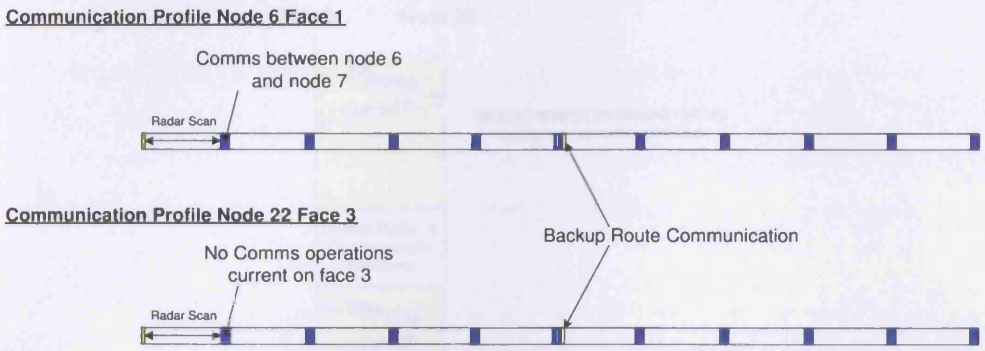



Figure 7.4: Backup link communications every 500ms

During backup route communication, simple messages would need to be sent. These would include at least the messages listed in Table 7.1. These messages allow for basic backup route maintenance and primary route establishment.

| Message | Message Description |
|---|---|
| Backup Ack | Acknowledge that the backup route is in normal operational state |
| Primary Req | Request backup route on current node be made into a primary route |
| Primary Ack | Acknowledge request for primary route initialization |
| Primary Dsq | Refuse promotion of backup route to primary route |
| Backup Rel | Release backup route and timeslot |
| Backup Rel Ack | Acknowledge release of backup route and timeslot |

Table 7.1: Backup route messages

Consider the message passing taking place in figure 7.3. During normal operation node 6 simply sends *Backup Ack* messages to node 6 to check that the backup route is operational. Node 22 then sends *Backup Ack* messages back to node 6 to confirm that the route is operational. When node 7 fails as in figure 7.3 the network becomes disconnected. Node 6 has no primary route on face 1 and therefore requests the promotion of its backup route to a primary route by sending a *Primary Req* as indicated in figure 7.5. If a primary route is available the node will return a *Primary Ack*. Node 6 then informs node 22 of the promotion of backup link using a *Backup Rel* message and backup route is removed and timeslot unallocated. Node 22 acknowledges this, removes the backup route and sends *Backup Rel Ack* and promotes the primary link. Both nodes are then able to communicate and in this case the network is reconnected. Backup routes can then be recalculated.

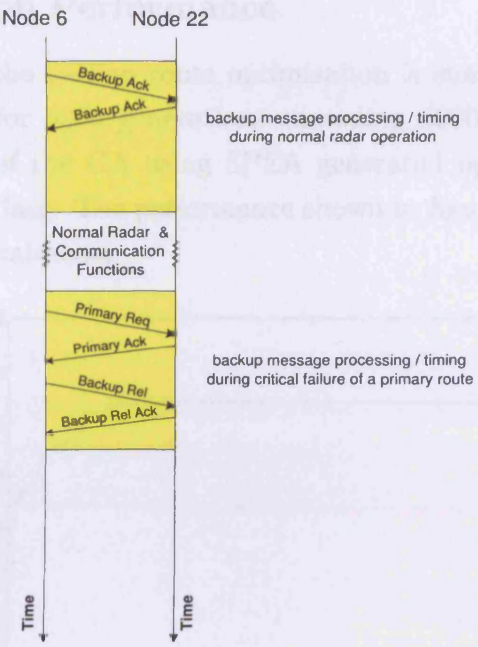



Figure 7.5: Backup route message processing and timing

### 7.2.4 Backup Route Placement in Netted Radar Networks

Backup route placement should be used to improve the robustness of the network, which for this thesis means that the connectivity should be improved as much as possible. In previous chapters connectivity was not used as a driving factor for the network design optimisation as it was time intensive. However the original network design could be completed before the backup routes are enabled. This would allow additional seconds for the backup route design process. The initial design could then be completed, implemented and the network initialised into an operational state before the backup route optimisation begins.

### 7.2.5 Design

The backup route design process is essentially the same as the connectivity only optimisation in section 4.10, with two modifications:

1. ***Optimisation starting procedure-*** The starting solution (input into the procedure) is the optimised network design found from design procedure. We call this the *base network.*
2. ***Repair operator-*** The repair operator ensures that the base network remains in the string representation.

These two modifications ensure that the base network remains central to the backup route design.

### 7.2.6 Optimisation Performance

The performance of the backup route optimisation is similar to that in section 4.10 with design times for 5000 generations exceeding 8000 seconds. Figure 7.6 shows the performance of the GA using SPEA generated optimised test network graph 13 with 1 link per face. The performance shown in figure 7.6 is typical of the backup route design optimisation.

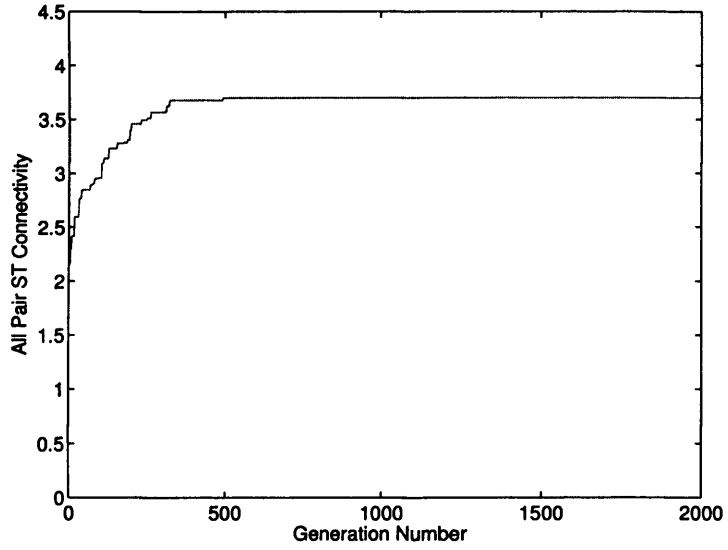



Figure 7.6: Graph indicating the performance of the GA starting with SPEA test network 13 with 1 link per face

The optimisation converges relatively quickly within 400-600 generations,taking 800-1000 seconds. While these times would be unacceptable in the network design process itself, they would be acceptable for backup routes. Node movement would limit the effectiveness of the backup route design but as node movements are infrequent this does not have a significant impact on the backup route design.

Another approach would be to apply new backup routes incrementally as the backup route optimisation continues to run. For example in figure 7.6 the best backup solution could be distributed/applied to the radar nodes every 30 seconds and the new backup routes initialised. Also, the connectivity increases rapidly within the first hundred generations as shown in figure 7.6. This results in a 50% increase in all pairs ST connectivity within the first 2 minutes. Therefore applying the backup network optimisation solutions incrementally would improve connectivity significantly.

### 7.2.7 Illustration

The following example shows the SPEA designed network 13 with 1 link per face. After the initial network design produced by the SPEA, the network consists solely of primary routes. This network is then passed to a secondary optimisation process as described in section 7.2.5. The network is then re-encoded into the GA string and used for the connectivity-only optimisation. This base string then remains unchanged throughout the GA to ensure that the original network remains intact and unaltered. The GA then adds additional links to improve connectivity, resulting in a final network design that is optimised for connectivity. After this connectivity optimisation process has been completed, the backup routes can then be distributed and set up by the individual nodes. The backup routes in figure 7.7 are displayed in red.

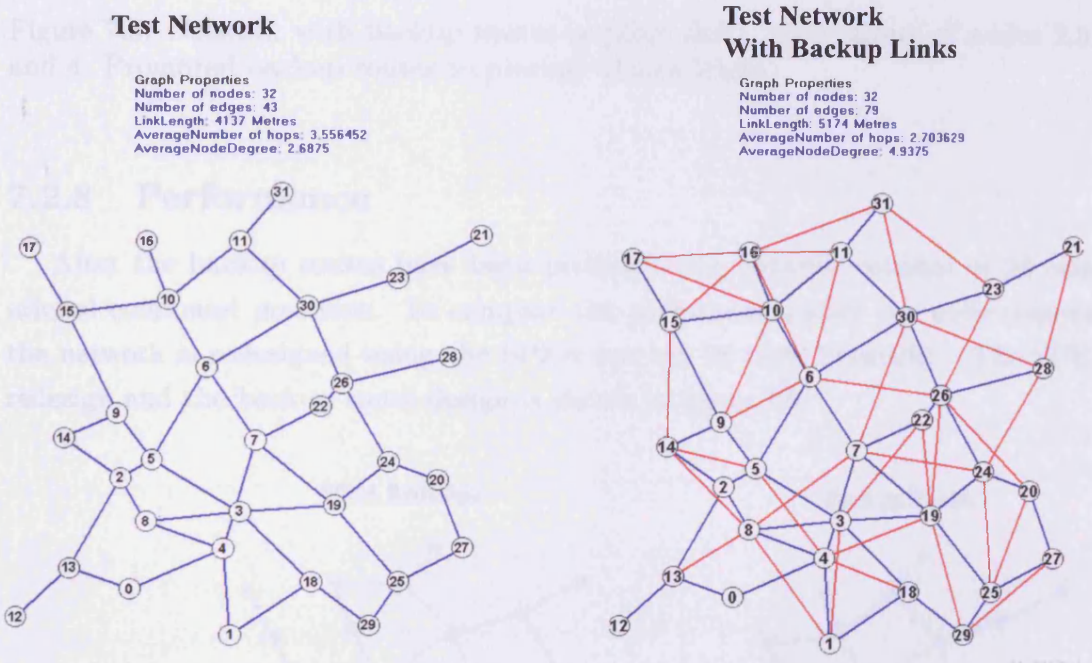



Figure 7.7: SPEA designed network with 1 link per face (left) and resulting network after backup routes have been calculated (right)

After critical failures, backup routes can quickly be promoted to primary links. An example of this is shown in figure 7.8 page 164. Nodes 2,5,3 and 4 are simultaneously destroyed. This results in irretrievable data loss at these nodes. Operational nodes using routes through the destroyed nodes must now route information using other available routes. For example, messages from node 1 to 14 would incur large delays routing around the destroyed nodes. This would amount to a minimum of 9 node hops (The path required would be similar to 1-18-29-25-19-7-6-10-15-9-14). After a period of approximately 500ms the nodes with failed primary routes would carry out the backup link promotion procedure as set out in section 7.2.3. This would enable 5 new primary routes. Now messages from node 1 to 14 can be routed

via node 8 in 2 hops. The disconnected section of the network (nodes 12, 13, 0 and 8) is also reconnected.

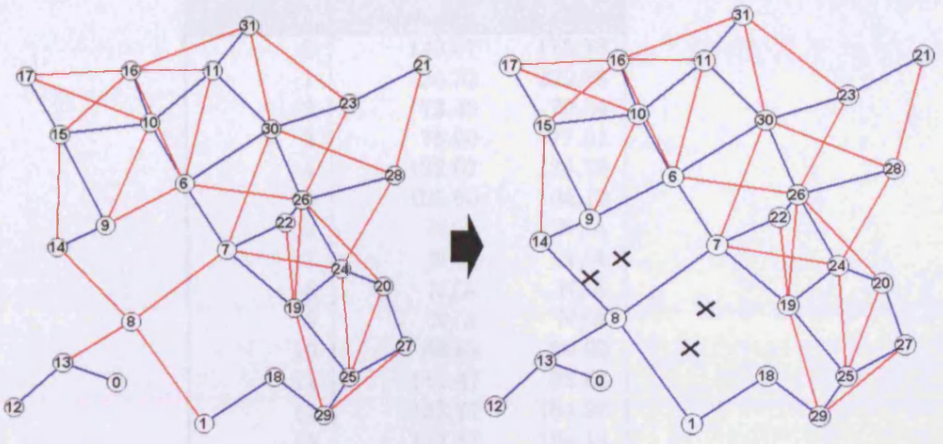


Figure 7.8: Network with backup routes in place (left), after failure of nodes 2,5,3 and 4. Promoted backup routes to primary routes (right)

### 7.2.8 Performance

After the backup routes have been promoted the network consists of 28 operational command positions. To compare the performance after the node removal, the network is redesigned using the SPEA (on the 28 node network) . The SPEA redesign and the backup route design is shown in figure 7.9.

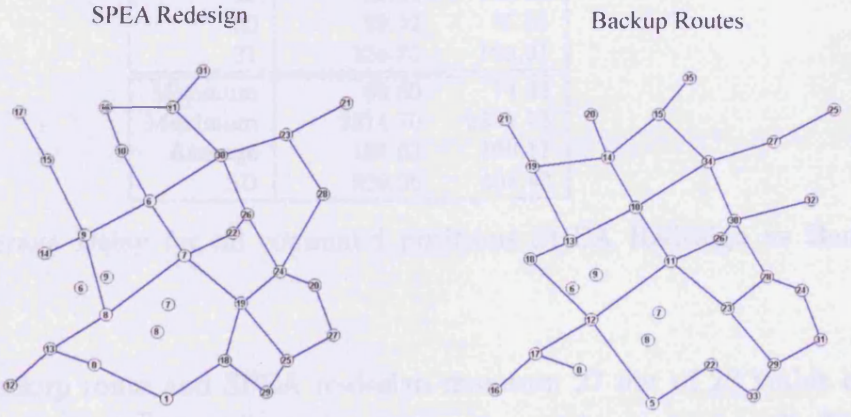



Figure 7.9: SPEA redesign after node removal (left) backup route for comparison (right)

Table 7.2 below shows a comparison of both networks

| Comm ID | Avg Msg delay(ms) SPEA Re-Design | Backup Routes |
|---|---|---|
| 0 | 149.07 | 175.78 |
| 1 | 124.32 | 229.88 |
| 2 | 73.49 | 79.54 |
| 3 | 75.60 | 77.01 |
| 4 | 122.07 | 134.78 |
| 5 | 105.60 | 94.12 |
| 6 | N/A | N/A |
| 7 | N/A | N/A |
| 8 | N/A | N/A |
| 9 | N/A | N/A |
| 10 | 86.82 | 90.00 |
| 11 | 112.47 | 92.84 |
| 12 | 133.17 | 184.25 |
| 13 | 117.58 | 194.14 |
| 14 | 117.02 | 112.20 |
| 15 | 163.20 | 125.96 |
| 16 | 90.03 | 99.54 |
| 17 | 168.70 | 145.24 |
| 18 | 94.27 | 160.36 |
| 19 | 81.38 | 90.58 |
| 20 | 81.42 | 85.38 |
| 21 | 2214.70 | 2232.73 |
| 22 | 71.74 | 77.03 |
| 23 | 110.45 | 192.77 |
| 24 | 76.04 | 87.36 |
| 25 | 104.92 | 105.84 |
| 26 | 69.80 | 74.34 |
| 27 | 92.57 | 94.05 |
| 28 | 135.98 | 155.09 |
| 29 | 99.10 | 111.10 |
| 30 | 89.42 | 85.81 |
| 31 | 124.75 | 103.31 |
| Minimum | 69.80 | 74.34 |
| Maximum | 2214.70 | 2232.73 |
| Average | 181.63 | 196.11 |
| SD | 399.36 | 401.43 |

Table 7.2: Average Delay for all command positions SPEA Redesign vs Backup Routes

Both the backup route and SPEA re-design maintain 27 out of 28 viable command positions. SPEA does have better average message delay times overall. Therefore it would seem an acceptable interim step to use instant backup routes while the SPEA re-design is calculated. To avoid too many network reconfigurations, the new backup routes can also be calculated while the old backup routes maintain the network.

### 7.2.9 Backup Route Connectivity Results

The backup route design process was tested on thirty SPEA designed networks with 1 link per face. The original optimised base network from the SPEA is used as the base/backbone network for the backup route optimisation. Optimisation is run for 1000 generations only.

| | All Pairs ST Connectivity | | | Time Taken |
|---|---|---|---|---|
| | | SPEA | | 1000 |
| Graph No. | SPEA 1 Link/Face | 1 Link/Face With Backup | Percentage Increase | Generations (seconds) |
| 1 | 1.98 | 2.91 | 47.10% | 1765 |
| 2 | 1.70 | 2.99 | 76.00% | 1786 |
| 3 | 1.51 | 2.52 | 66.60% | 1845 |
| 4 | 1.84 | 2.94 | 59.40% | 1862 |
| 5 | 1.80 | 2.99 | 65.60% | 1749 |
| 6 | 1.10 | 1.76 | 59.30% | 1659 |
| 7 | 1.92 | 2.90 | 51.40% | 1668 |
| 8 | 1.65 | 2.62 | 58.70% | 1759 |
| 9 | 1.72 | 3.14 | 82.70% | 1845 |
| 10 | 2.07 | 3.64 | 75.60% | 1692 |
| 11 | 2.09 | 3.81 | 82.70% | 1688 |
| 12 | 1.96 | 3.61 | 84.50% | 1599 |
| 13 | 2.02 | 3.41 | 68.40% | 1785 |
| 14 | 1.90 | 3.69 | 94.40% | 1722 |
| 15 | 2.06 | 3.92 | 90.60% | 1739 |
| 16 | 1.91 | 3.75 | 96.80% | 1695 |
| 17 | 2.04 | 3.85 | 89.30% | 1786 |
| 18 | 2.23 | 4.12 | 84.80% | 1878 |
| 19 | 1.79 | 2.96 | 65.60% | 1826 |
| 20 | 1.85 | 3.52 | 89.90% | 1855 |
| 21 | 1.63 | 2.58 | 58.30% | 1769 |
| 22 | 1.51 | 2.58 | 71.20% | 1902 |
| 23 | 1.94 | 3.92 | 102.40% | 1648 |
| 24 | 1.34 | 2.70 | 101.90% | 1759 |
| 25 | 1.65 | 3.03 | 83.90% | 1735 |
| 26 | 1.30 | 3.12 | 139.50% | 1762 |
| 27 | 1.88 | 3.20 | 70.40% | 1726 |
| 28 | 1.56 | 2.95 | 89.40% | 1698 |
| 29 | 1.70 | 2.80 | 64.60% | 1810 |
| 30 | 2.14 | 3.31 | 54.60% | 1879 |
| Minimum | 1.10 | 1.76 | 47.10% | 1599 |
| Maximum | 2.23 | 4.12 | 139.50% | 1902 |
| Average | 1.79 | 3.16 | 78.51% | 1762 |
| SD | 0.29 | 0.59 | 22.02% | 82 |

Table 7.3: Backup route results for SPEA designed base networks with 1 link per face

The results indicate substantial increases in connectivity. All networks achieve between 47% and 139% increase in connectivity with an average improvement of 78%. The optimisation times for 1000 generations range between 1599 and 1902 seconds.

It is also important to note that the connectivity values are above the 2 links per face SPEA designed networks as shown in Table 7.4. Backup route designs have connectivity values between 1% and 15% greater than the equivalent SPEA designed network with 2 links per face.

| | All Pairs ST Connectivity | | |
|---|---|---|---|
| Graph No. | SPEA 1 Link/Face With Backup | SPEA 2 Link/Face | Percentage Increase |
| 1 | 2.91 | 2.87 | 1.60% |
| 2 | 2.99 | 2.88 | 3.90% |
| 3 | 2.52 | 2.31 | 8.90% |
| 4 | 2.94 | 2.83 | 4.00% |
| 5 | 2.99 | 2.88 | 3.50% |
| 6 | 1.76 | 1.66 | 5.70% |
| 7 | 2.90 | 2.80 | 3.40% |
| 8 | 2.62 | 2.56 | 2.20% |
| 9 | 3.14 | 3.08 | 2.10% |
| 10 | 3.64 | 3.58 | 1.60% |
| 11 | 3.81 | 3.40 | 12.00% |
| 12 | 3.61 | 3.30 | 9.40% |
| 13 | 3.41 | 3.24 | 5.20% |
| 14 | 3.69 | 3.51 | 5.30% |
| 15 | 3.92 | 3.65 | 7.30% |
| 16 | 3.75 | 3.69 | 1.60% |
| 17 | 3.85 | 3.75 | 2.70% |
| 18 | 4.12 | 4.09 | 0.70% |
| 19 | 2.96 | 2.78 | 6.30% |
| 20 | 3.52 | 3.39 | 3.90% |
| 21 | 2.58 | 2.54 | 1.30% |
| 22 | 2.58 | 2.23 | 15.80% |
| 23 | 3.92 | 3.49 | 12.20% |
| 24 | 2.70 | 2.59 | 4.10% |
| 25 | 3.03 | 2.98 | 1.50% |
| 26 | 3.12 | 2.87 | 9.00% |
| 27 | 3.20 | 3.05 | 5.10% |
| 28 | 2.95 | 2.67 | 10.50% |
| 29 | 2.80 | 2.63 | 6.70% |
| 30 | 3.31 | 3.13 | 5.70% |
| Minimum | 1.76 | 1.66 | 0.70% |
| Maximum | 4.12 | 4.09 | 15.80% |
| Average | 3.16 | 3.01 | 5.62% |
| SD | 0.59 | 0.58 | 4.12% |

Table 7.4: Backup route results for networks designed with SPEA, 1 link/face and backup routes compared with SPEA designed with 2 links/face.

### 7.2.10 Summary

Using backup routes clearly allows 1 link per face network designs to be implemented but with the benefit of higher all pairs ST connectivity. The connectivity values are in fact higher than SPEA designed networks with 2 links per face. The fail over performance of the links is dependent on the number of backup timeslots which has been limited to a single timeslot every 500ms. This could be increased but either communication time or radar time would have to be decreased to accommodate the additional backup timeslots. Although the timeslot would be extremely small, factors such as beam switching and link setup time would have to be taken into account. This more comprehensive study can only be done if both radar function and radar characteristics are simulated. This radar simulation is considered beyond the scope of this project.

The backup route design as described is inherently global. This allows high connectivity values to be achieved. A localised version would attain similar connectivity values to the DARTC designed networks with 2 links per face. A hybrid approach might be to use the second phase of the 2 links per face DARTC algorithm to construct backup routes immediately and the global algorithm could then be used once good solutions have been generated by the optimisation. This method would benefit from the distributed initial backup routes calculated almost immediately followed by the routes designed globally as the optimisation progresses.

CHAPTER

# EIGHT

## Node Movement

When nodes are removed from the network through failure, the network can become disconnected, in which case the network would have to be redesigned. This cannot be avoided unless procedures such as backup route design in section 7.2 are implemented. As the nodes in this thesis are considered not able to communicate during movement, movement can be considered as a node disconnection followed by a node addition at some different location. Having already briefly dealt with how node deletions might be handled in section 7, this section will focus on the additions of nodes into the network. This section will consider two types of node additions:

1. Multiple node additions to extend coverage.
2. Single node additions.

## 8.1 Multiple Node Additions

When radar coverage needs to be extended, radars could be added in bands at any edge of the network. This would involve adding multiple nodes into the network. An example is shown in figure 8.1 page 170. Optimised test network 13 can be extended to the East by adding nodes as shown. The six nodes (coloured black) are added in a non-random manner to extend coverage. The extended coverage is shown by the red overlay.

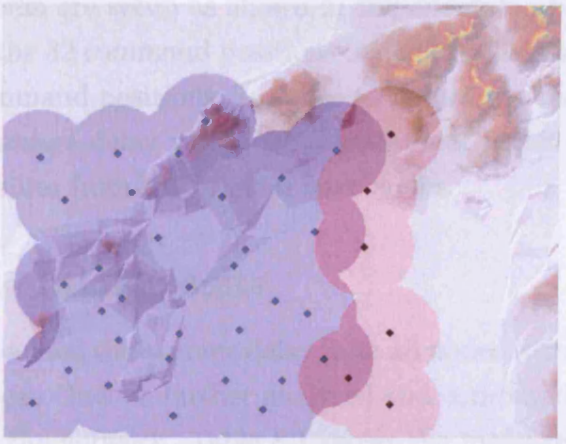


Figure 8.1: Multiple Node Additions

Multiple node additions can be integrated into the network either through a complete redesign of the network or a local approach such as that employed by DARTC.

### 8.1.1 Local Addition method

The first method tested was local node addition. The original 32 node SPEA designed networks with 1 link per face were taken as the base networks. Nodes were then added in a non-random manner to the East of the networks to extend network coverage. An example of the node addition is shown in figure 8.2, black nodes being new nodes. The new nodes join the network by running the DARTC algorithm for 1 link per face only on each of the added nodes.

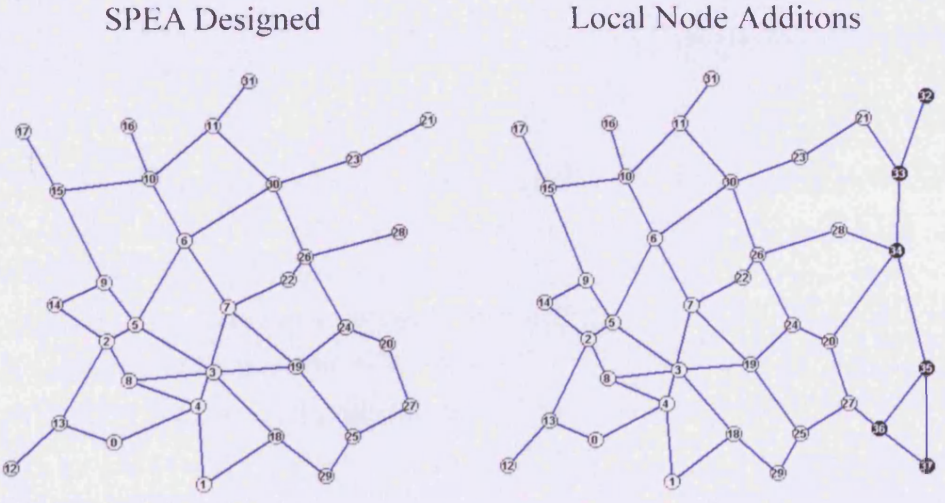



Figure 8.2: Local node additions

All 30 test networks are setup as shown in the example above. Simulations are then carried out for the 32 command positions in the original network. Added nodes are not tested as command positions. In order to determine the performance of the added nodes, the message delay values from additional nodes can be compared to the message delay values from the original base nodes.

### 8.1.2 Local Addition results

When new nodes are added the average delay from all nodes is tracked. The messages from the new nodes can then be further analysed and compared to other nodes that were already within the network. Table 8.1 shows the message delay results of the multiple node additions with test graph 13 SPEA 1LPF. Messages from the newly added nodes only are shown on the first column, messages from base nodes (nodes that existed before the new nodes were added) are shown in the second column and the third column shows the original message delays without the node additions.

| Comm ID | Avg msg delay(ms) | | |
|---|---|---|---|
| | New Nodes | Base Nodes | Previous Base |
| 0 | 330.79 | 186.47 | 183.96 |
| 1 | 137.58 | 81.03 | 96.99 |
| 2 | 140.09 | 85.30 | 76.79 |
| 3 | 156.97 | 105.51 | 74.85 |
| 4 | 165.97 | 108.89 | 95.75 |
| 5 | 126.05 | 74.27 | 76.22 |
| 6 | 122.96 | 71.10 | 65.77 |
| 7 | 147.12 | 92.19 | 65.78 |
| 8 | 162.77 | 91.83 | 80.23 |
| 9 | 132.47 | 84.67 | 78.51 |
| 10 | 129.21 | 79.29 | 77.28 |
| 11 | 119.86 | 89.76 | 96.49 |
| 12 | 144.61 | 139.45 | 136.75 |
| 13 | 128.76 | 122.07 | 122.50 |
| 14 | 87.67 | 82.35 | 81.38 |
| 15 | 132.66 | 129.66 | 126.38 |
| 16 | 142.77 | 139.40 | 88.93 |
| 17 | 161.85 | 157.19 | 150.21 |
| 18 | 115.56 | 85.70 | 95.85 |
| 19 | 126.48 | 98.93 | 72.04 |
| 20 | 80.86 | 87.73 | 75.37 |
| 21 | 158.39 | 93.30 | 10246.90 |
| 22 | 1762.01 | 736.63 | 67.82 |
| 23 | 10956.87 | 4590.99 | 8821.17 |
| 24 | 159.42 | 99.92 | 70.86 |
| 25 | 94.99 | 91.64 | 97.70 |
| 26 | 73.89 | 68.71 | 67.89 |
| 27 | 106.17 | 101.32 | 87.72 |
| 28 | 3701.18 | 2133.49 | 26228.00 |
| 29 | 136.32 | 98.93 | 90.35 |
| 30 | 101.23 | 93.49 | 86.22 |
| 31 | 122.46 | 124.93 | 110.39 |
| Minimum | 73.89 | 71.10 | 65.77 |
| Maximum | 10956.87 | 4590.99 | 26228.00 |
| Average | 636.44 | 443.77 | 2184.91 |
| SD | 2003.93 | 1096.54 | 6409.08 |

Table 8.1: Average delay values for new nodes compared to base nodes and previous simulations *Test network 13*

Figure 8.3 is a graphical representation of Table 8.1. What is most significant is that the group of new nodes tends to have higher average message delay values than other nodes. In this particular example there is only a single node where the additional nodes have caused unacceptable delays to occur (node 22) where previously the command position was viable. The difficult command positions still remain, even with the new node additions.

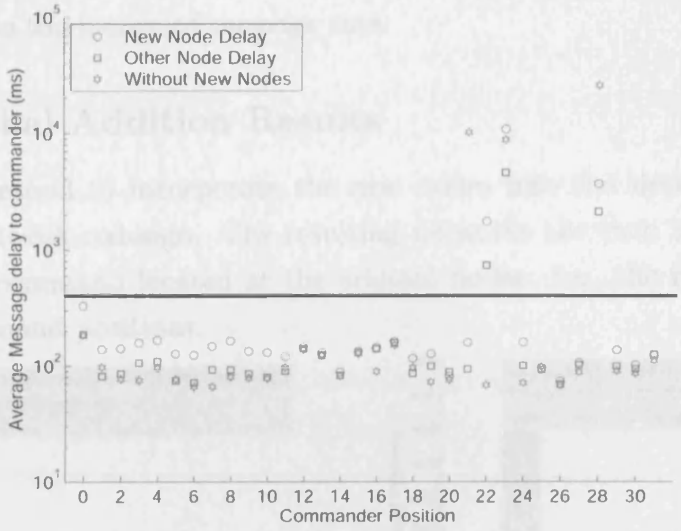



Figure 8.3: Local Node Additions

To evaluate the effect the new nodes have on the underlying base network, the message delays to the commander from the new nodes are compared to the underlying base network node delays (Nodes that existed before the new nodes were added). Figure 8.4 shows the message delay ranges for messages received by the commander from the newly added nodes for all simulations. Figure 8.5 shows the message delay ranges to the commander from the original nodes in the network.

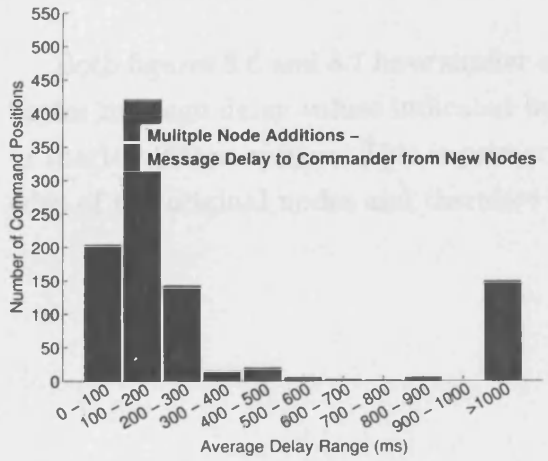



Figure 8.4: Commander message delays from new nodes

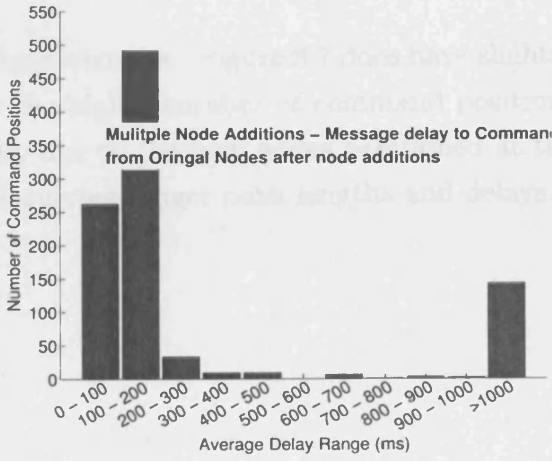



Figure 8.5: Commander message delays from original nodes after node addition

The message delays to the commander from the group of new nodes are greater than the delays from the original base nodes after the new nodes have been added. This is shown by the greater frequency of messages in the 200-300ms range from the new nodes. Although the number of messages in the 200-300ms range have increased, the new nodes do not significantly alter the shape of the histogram i.e 8.4 compared to figure 5.19. The number of command positions that have unacceptable delays has

remained almost unchanged with only a further 8 node positions overall becoming unusable due to the increased message rate.

### 8.1.3 Global Addition Results

The second method to incorporate the new nodes into the network is to initiate a complete network redesign. The resulting networks are then tested for message delays to the command located at the original nodes. i.e. the new nodes are not tested as command positions.

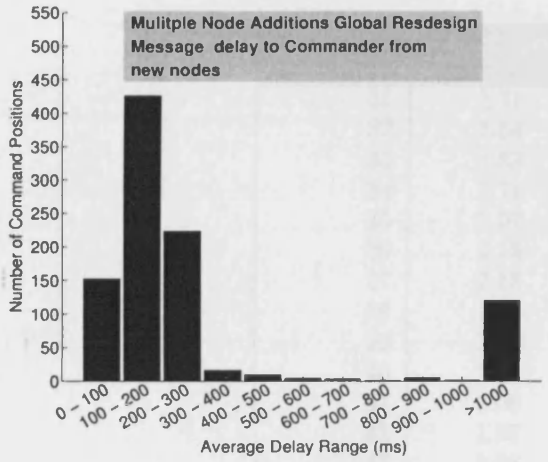



Figure 8.6: Commander message delays from new nodes

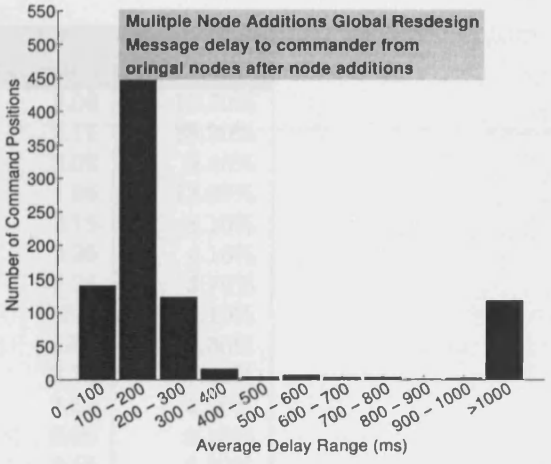



Figure 8.7: Commander message delays from original nodes after node addition

Both figures 8.6 and 8.7 have similar characteristics. Figure 8.7 does have slightly better message delay values indicated by the higher number of command positions in the 100-200ms ranges. This is primarily due to the new nodes positioned at the edge of the original nodes and therefore incurring longer path lengths and delays.

### 8.1.4 Connectivity Comparison of Local Vs Global Design Connectivity

Table 8.2 shows the all pairs ST-connectivity for multiple node additions (with node additions, the graphs are numbered 31-60). Global re-design (using SPEA) can offer significant increases in connectivity ranging from 3-39% and averaging 15% increase over the local design DARTC method. This is evident in Table 8.2 where the connectivity of the global re-design method is always above the local counterpart. The global re-design is on average 15.75% higher than the local addition.

| Graph No. | Local Addition | Global Re-Design | Percentage Increase |
|---|---|---|---|
| 31 | 1.71 | 2.03 | 15.50% |
| 32 | 1.54 | 2.15 | 28.20% |
| 33 | 1.83 | 2.02 | 9.40% |
| 34 | 1.71 | 1.96 | 13.00% |
| 35 | 1.97 | 2.15 | 8.10% |
| 36 | 2.15 | 2.25 | 4.10% |
| 37 | 2.17 | 2.25 | 3.70% |
| 38 | 1.62 | 1.78 | 9.10% |
| 39 | 1.53 | 1.79 | 14.30% |
| 40 | 1.98 | 2.14 | 7.40% |
| 41 | 2.06 | 2.13 | 3.30% |
| 42 | 1.87 | 2.06 | 9.10% |
| 43 | 2.03 | 2.15 | 5.30% |
| 44 | 2.08 | 2.29 | 9.10% |
| 45 | 1.60 | 2.19 | 27.30% |
| 46 | 1.36 | 2.22 | 38.60% |
| 47 | 1.85 | 2.10 | 11.90% |
| 48 | 2.09 | 2.20 | 5.10% |
| 49 | 1.95 | 2.20 | 11.50% |
| 50 | 1.61 | 2.17 | 25.90% |
| 51 | 1.47 | 1.64 | 10.60% |
| 52 | 1.32 | 1.94 | 31.70% |
| 53 | 2.05 | 2.17 | 5.50% |
| 54 | 1.27 | 1.83 | 30.50% |
| 55 | 1.47 | 2.04 | 27.80% |
| 56 | 1.26 | 1.81 | 30.80% |
| 57 | 1.82 | 2.12 | 14.10% |
| 58 | 1.61 | 2.07 | 21.90% |
| 59 | 1.85 | 1.90 | 2.30% |
| 60 | 1.79 | 2.45 | 27.00% |
| Minimum | 1.26 | 1.78 | 3.30% |
| Maximum | 2.17 | 2.45 | 38.60% |
| Average | 1.75 | 2.07 | 15.75% |
| SD | 0.27 | 0.19 | 11.06% |

Table 8.2: Multiple node addition connectivity comparison of local vs global methods

## 8.2 Single Node Additions - Local Addition

Single node additions are new single radar nodes positioned within range of the radar network. This can be at the perimeter/boundary of the radar network or at some location within the radar network. We also assume that the single node addition occurs after the main network is fully operational and has a complete network design in place. Local node addition is therefore concerned with joining the network based on local decisions and therefore provides minimal disruption to the network.

There are two different possible scenarios that could occur when new nodes join the network.

1. The new node is within range of existing nodes with an unused communicating face.
2. The new node is within range of existing nodes but there are no spare communicating faces.

Figure 8.8 shows scenario 1 where the single node is within range of a node that has an unused communicating face. In this case node 22 is within range and it is simply a case of the new node (node 32) adding a single link to node 22 as shown in figure 8.9.

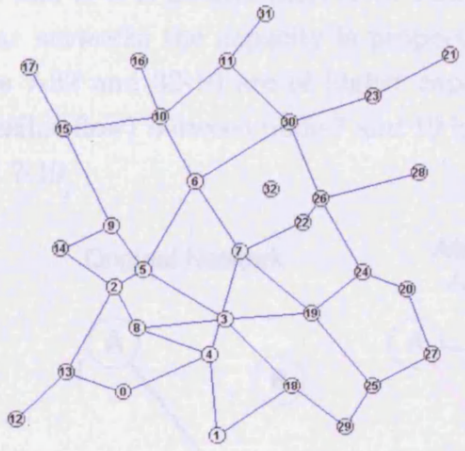


Figure 8.8: Single node addition Scenario 1

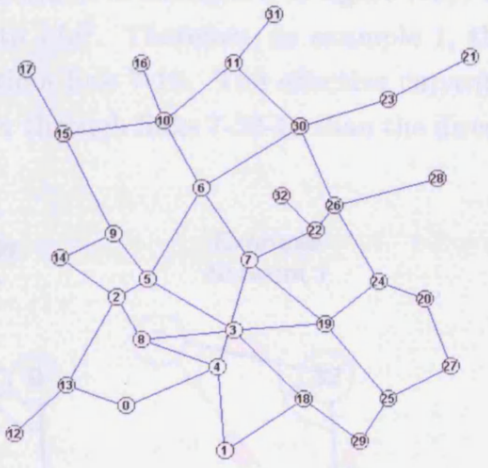


Figure 8.9: Single node addition Scenario 1 with 1 new link added

Figure 8.11 shows scenario 2 where the single node is not in range of any nodes with unused communicating faces. In this case, one or more links need to be broken in order for the new node (node 32) to be added to the network. Links 7-22 and 7-19 are broken and new links added to the new node as shown in figure 8.12. A simple heuristic can be used to ensure that good links are chosen. The local addition algorithm is shown on the following page. When breaking links to allow new links to the new nodes, simple geometric properties can be utilised to ensure that good links are chosen.

Consider the example shown in figure 8.10. The red crosses indicate the radar faces and the red dashed line is the original link. In the leftmost figure, node B requires the link between A and C to be broken as node B can only connect to nodes A or B by connecting to one of the already used faces. Therefore the link is broken and new links between A-B and B-C are formed. As each radar face covers 90 degrees and by ensuring the angle $\angle ABC$ is more than 90 degrees, then the links connecting A-B and B-C are guaranteed to be on different faces. In this case node B has links on faces 1 and 3.

As the angle $\angle ABC$ is selected to be greater than 90 degrees, both links A-B and B-C are guaranteed to be shorter than A-B. This means that the capacity between A-B and B-C is greater than A-C. This is illustrated in example 1 in figure 8.10. In radar networks the capacity is proportional to $1/d^2$. Therefore, in example 1, the links 7-32 and 32-19 are of higher capacity than link 7-19. The effective capacity (possible flow) between node 7 and 19 is higher through links 7-32-19 than the direct link 7-19.

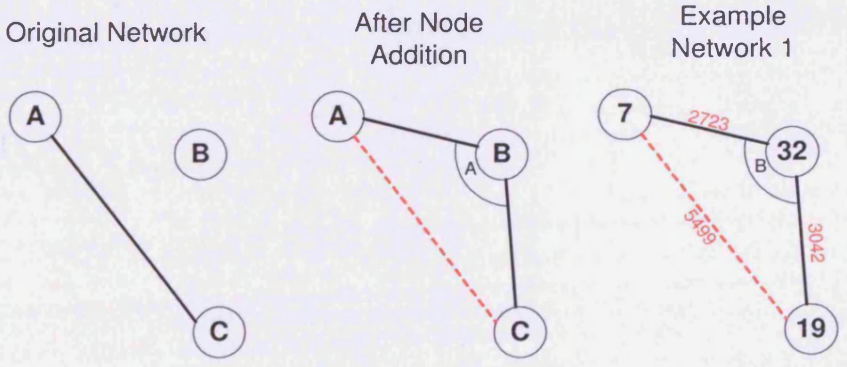



Figure 8.10: Local node addition

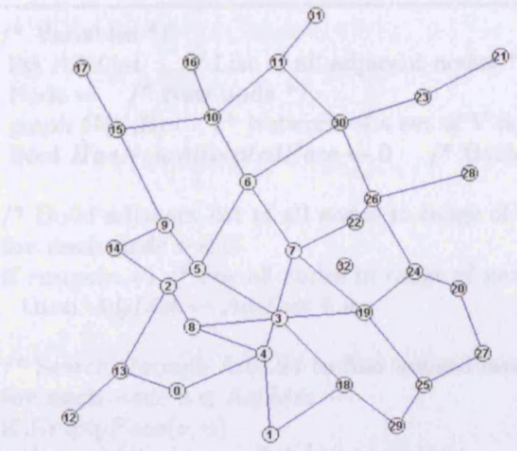


Figure 8.11: Single node addition Scenario 2

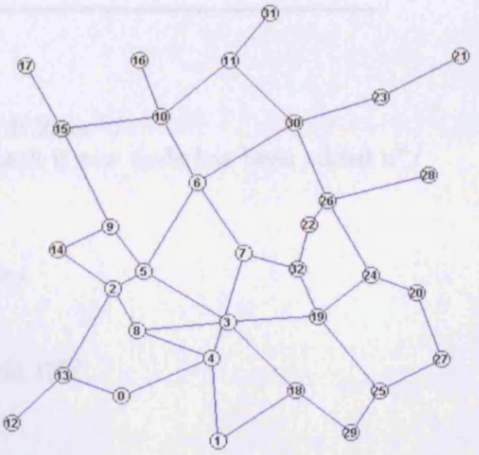


Figure 8.12: Single node addition Scenario 2 with 2 new links

### 8.2.1 Local Single Node Addition Algorithm

The algorithm below details the node addition for both scenarios described in section 8.2.

**Algorithm 8.2.1:** NEW NODE ADDITION($n$)

```
/* Variables */
 list AdjList    /* List of all adjacent nodes */
 Node n    /* New node */
 graph G(V,E)     /* Network of a set of V nodes and E links */
 Bool HasNewAllocatedFace ← 0    /* Boolean to check if new node has been added n*/

/* Build adjacent list of all nodes in range of n */
for each node v ∈ G
if range(n,v) /* For all nodes in range of new node n*/
  then AdjList ← AdjList + v

/* Search through AdjList to find unused face (Scenario 1)*/
for each node v ∈ AdjList
if EmptyFace(v,n)
  then Addlink(v,n)/* Add new link */
       HasNewAllocatedFace ← 1

/* If nodes in range have no unused communicating faces (Scenario 2) */
if HasNewAllocatedFace
for each node v ∈ AdjList
    for each node u ∈ AdjList
        if LinkExists(u,v) And GetAngle(u,n,v) > 90
           RemoveLink(u,v)
           Addlink(v,n)
           if RemoveConflicts(n)
             HasNewAllocatedFace ← 1

/* If no nodes within range or acceptable faces */
if HasNewAllocatedFace
    RaiseError/* nodes cannot be added */
```

***Functions:***

$range(u,v)$ ← Checks if node v is within communication range of u
$EmptyFace(u,c)$ ← Determines if u and c have unused faces in the direction of each other
$CorrFace((u,n,v))$ ← Checks if u n and n v have correct opposing faces
$Addlink(u,t)$ ←Adds Link between node u and node t
$LinkExists(u,v)$ ←Determines if there is a link between u,v
$GetAngle(u,n,v)$ ←Determines the angle ∠u,n,v
$RemoveLink(u,v)$ ←Removes Link between u and v
$RemoveConflicts(n)$ ←Removes conflicts after new links are added. If faces have more than maximum number of links, a link is removed at random from one of these faces (returns 1 if all conflicts have been resolved)

### 8.2.2 Local Addition Results

When new nodes are added to the network the number of messages increases due to the additional radar coverage of the additional node. This changes the total number of messages and could potentially change the dynamics of the network. This additional congestion could cause message delays to become considerable and the commander position to become unusable.

1000 single nodes are added at completely random positions to any of the 30 SPEA designed random test networks with 1 link per face i.e. one new node is added at a random position to a randomly chosen network from the 30 test networks and this process is then repeated 1000 times. The new node is then added to the network using the algorithm described in section 8.2.1 and compared to the complete SPEA network redesign. A command position is then chosen at random from this network (excluding the new node). If the new node enables a previously unviable command position to become viable, this is recorded. The critical aspect to node addition is whether the message delays can remain below the threshold of 500ms with the resulting increase in traffic.

Table 8.3 shows the results for 1000 tests.

| Description | Local Addition | SPEA Re-Design |
|---|---|---|
| Total No. of Node Additions | 1000 | |
| Total No. of Boundary Node Additions | 245 | |
| Total No. of Internal Node Additions | 715 | |
| Total No. of Unusable Node Additions (out of Range) (see below) | 40 | |
| Total No. of Known Viable Command Positions Tested (see below) | 909 | |
| Total No. of Viable message delays after node addition | **901** | **905** |
| Total No. of New viable command positions | **8** | **9** |
| Avg No. of additional messages per Additional Node | 209.7 | |
| Avg delay of Additional Messages (For Viable Command Positions) | 195.2ms | 202.4ms |

Table 8.3: Single Node Addition Results - Local

Note.

**Total No. of Unusable Node Additions (out of Range)** are nodes placed outside the range of any other node and so does not send any radar messages via the network.

**Total No. of Known Viable Command Positions Tested** are Command positions that incurred viable network delays without node additions are considered known viable command positions.

Of the 1000 command positions chosen, 909 were known to be viable command positions before the new node addition. With the local node addition algorithm, 901 command positions remained viable as opposed to 905 with a complete SPEA re-design.

Of the 91 networks which were previously unviable, 8 were viable after the local node addition and 9 with SPEA. This is due to either the new node providing an alternative route to relive congestion or improving throughput due to its proximity to the congested nodes.

## 8.3 Node Additions Summary

Multiple node additions to extend coverage do not have any significant effect on overall performance. Both local addition and global redesign methods were tested to include the new nodes and extend overall coverage. The resulting increase in traffic after multiple node additions did not cause significant congestion. Complete re-design did however increase the overall connectivity of the networks compared to the local addition equivalent.

Single node additions were tested with both a local addition algorithm and a complete redesign. The small number of additional messages resulting from the new node does not have any real influence on the performance of the network. The relatively high bandwidth of the networks easily handles the additional message load. Whilst it does demonstrate that single node additions can easily be implemented, a more appropriate analysis might be to make multiple single node additions throughout the network until performance deteriorates. This is a difficult and time consuming analysis and is beyond the scope of this initial study.

CHAPTER

# NINE

# Variable Timeslots

Traditional network design uses high bandwidth infrastructure depending on the requirements. Client/server architectures employ increasingly higher capacity infrastructure as the requirements and traffic to the server increase. For example, a network might implement fiber optic connections between switches/routers and servers as traffic increases[96]. In radar networks, the majority of messages are sent to the commander and obviously communication links nearer to the commander will be highly utilised or overloaded. This problem has been highlighted in previous chapters. The obvious solution would be to increase the power and hence bandwidth of these links. However it is assumed that the radar nodes in this thesis are not able to increase power in this manner. This section suggests an alternative.

## 9.1 Variable Timeslots Procedure

An alternative method is to increase the communications time of any radar node with additional traffic requirements. As very few nodes have overloaded links, it is expected that improving only these overloaded links would have a dramatic effect in reducing average message delay. The procedure is simple. If a node has a queue which exceeds the data-rate of a single timeslot, it extends the timeslot in an attempt to relieve the queue. This method ensures that each node only needs to know the state of its own queues.

Figure 9.4 shows how variable timeslots might be implemented. Figure 9.4 shows a partial section of a network. It is assumed that all messages to the commander have to traverse the communication link between nodes 1 and 2. If at some time, $T_1$, the number of messages in the queue exceeds that required to guarantee to the delivery to the next node within a single timeslot, both nodes agree to allocate additional time in communication mode. The maximum amount of time by which a timeslot can be extended is limited to 40ms. This guarantees that 50% of the radar

time is reserved for radar operations. If the queue is reduced to an acceptable level at time $T_2$ then the communications channel is closed and the radar operation can continue as normal. This ensures that the communications channel is not unnecessarily maintained.

This concept is highlighted in figure 9.1. It shows a 100ms time interval. The blue timeslot is the normal communication timeslot, the yellow is the possible time to which the communication timeslot can be extended in increments of 10ms. This results in a possible 50ms of communication time. By only allowing 40ms of additional communication time, 50ms of radar function time is guaranteed.

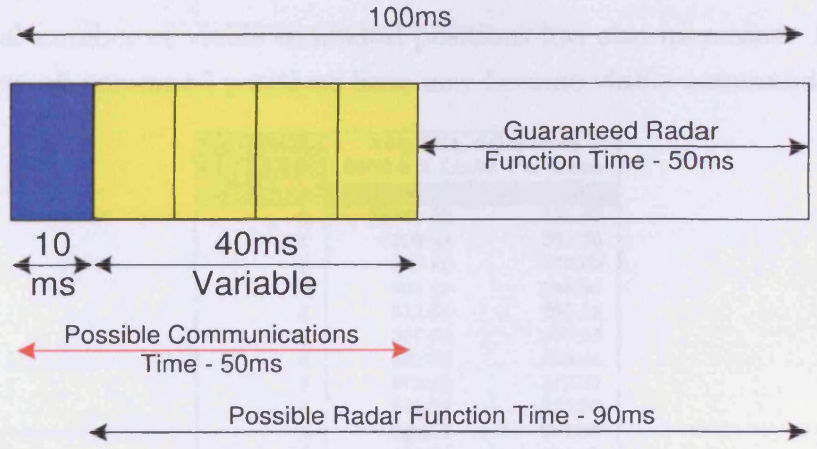



Figure 9.1: Variable Timeslots example

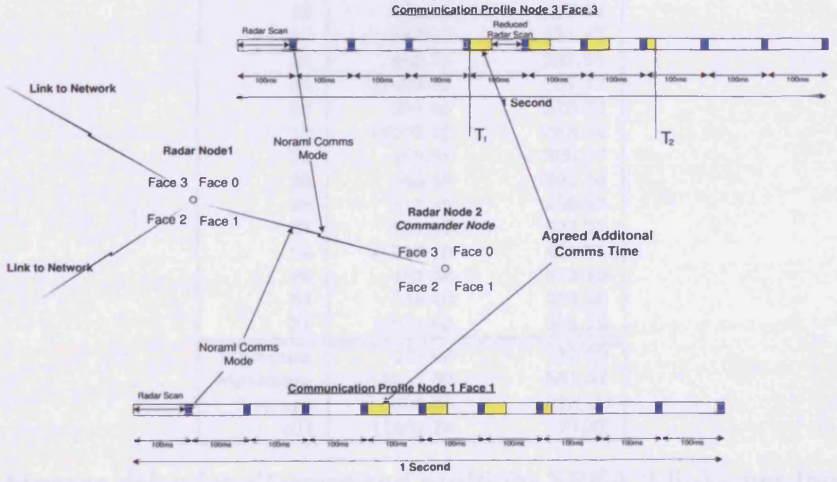



Figure 9.2: Variable Timeslots

In order that the radar function is not impaired at the most crucial times i.e. when a target is in range, no additional timeslots are allowed on the radars that are

currently in surveillance, tracking or engagement modes.

## 9.2 Worked Example

As an illustrative example consider the SPEA designed test network 13 (figure 5.13 page 130) with 2 links per face. Using variable timeslots (abbreviated to VT) drastically reduces the average delay times to the commander. The average delay time for test network 13 is reduced from 6305.85ms to 316.12ms. This can be attributed to the variable timeslots being able to compensate for the difficult command positions. This has the effect of dramatically lowering the overall average delay.

The total number of viable command positions has also increased. In table 9.1 shown below, all command positions have now become viable command positions.

| Comm ID | Avg Msg delay(ms) SPEA 2 Links Per Face | |
|---|---|---|
| | No VT | VT |
| 0 | 18744.50 | 335.62 |
| 1 | 3309.98 | 355.36 |
| 2 | 357.16 | 273.89 |
| 3 | 291.47 | 248.61 |
| 4 | 302.53 | 256.18 |
| 5 | 337.64 | 267.15 |
| 6 | 291.78 | 226.14 |
| 7 | 273.68 | 212.22 |
| 8 | 349.55 | 272.07 |
| 9 | 368.52 | 267.88 |
| 10 | 414.24 | 306.62 |
| 11 | 12945.50 | 352.26 |
| 12 | 14781.80 | 393.10 |
| 13 | 451.14 | 351.86 |
| 14 | 336.57 | 276.20 |
| 15 | 3956.47 | 288.86 |
| 16 | 485.63 | 356.92 |
| 17 | 32679.50 | 364.19 |
| 18 | 350.39 | 283.74 |
| 19 | 283.50 | 234.42 |
| 20 | 408.74 | 293.93 |
| 21 | 32499.20 | 581.22 |
| 22 | 307.46 | 228.72 |
| 23 | 16603.10 | 383.54 |
| 24 | 380.94 | 305.37 |
| 25 | 543.48 | 392.14 |
| 26 | 317.35 | 256.02 |
| 27 | 521.15 | 322.21 |
| 28 | 45017.60 | 481.75 |
| 29 | 485.38 | 329.19 |
| 30 | 359.40 | 253.28 |
| 31 | 13031.80 | 365.23 |
| Minimum | 273.68 | 212.22 |
| Maximum | 45017.60 | 581.22 |
| Average | 6305.85 | 316.12 |
| SD | 11487.78 | 77.57 |

Table 9.1: Average delay for all command positions SPEA 2 links per face vs SPEA 2 links per face with variable timeslots

### 9.2.1 Worked Example Test Network 13

Consider test network 13 SPEA designed with 2 links per face (Shown in figure 9.3 for reference purposes) and the commander located at position 21. Without variable timeslots the average message delay to the commander is 32499ms. With variable timeslots it is reduced to 581ms (from Table 9.1 page 184). This is accomplished by using an additional 3593(7186[1]) timeslots throughout the network. This does not have a significant overall effect on radar function. Table 9.2 summarises the timeslots requirements. The required overhead (where the overhead is the percentage of total radar time used by the variable timeslots) in this case is 0.25% which would reduce the radar function performance by 0.25%. This example is typical of the effect variable timeslots have on the radar and communication performance network wide.

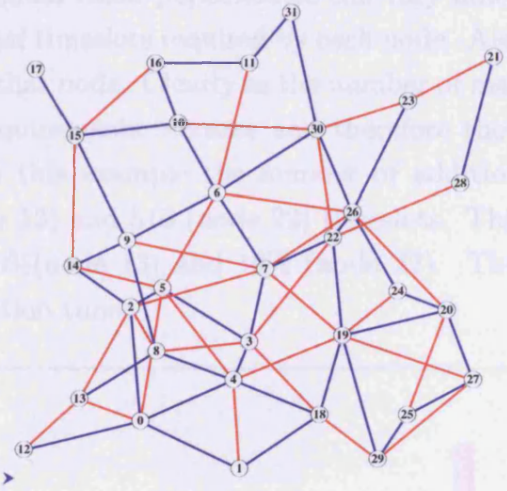



Figure 9.3: SPEA designed test network 13 with 2 links per face

| | |
|---|---|
| Total No. of Radar Function Timeslots | 1433600 |
| Total No. of Communication Timeslots | 143360 |
| Total No. of Additional Timeslots Used | 7186 (3593 x 2) |
| Total Additional Timeslot Overhead | 0.5% |

Table 9.2: Total variable timeslots required for command position 21

[1]Both nodes require the additional timeslots to be synchronised. Therefore each node requires additional timeslots

Total additional timeslots overhead is the percentage of total radar time required by the network. The simple calculations required for the above analysis are shown below:

$$
\begin{aligned}
\text{Number of Radar Faces} &= NRF \\
\text{Number of Nodes} &= NN \\
\text{Simulation Time} &= ST \\
\text{Total Number of Timeslots per Second} &= TNTS \\
\text{Total Radar Function Times} &= NRF \times NN \times ST \times TNTS \\
&= 4 \times 32 \times 112 \times 100 = 1433600 \\
\text{Total Additional Timeslot Overhead} &= ((3593 \times 2)/1433600) \times 100 = 0.5\%
\end{aligned}
$$

The effect on individual radar performance can vary however. Figure 9.4 shows the number of additional timeslots required by each node. Also shown is the number of messages routed by that node. Clearly as the number of messages routed increase, the communication requirements increase and therefore the number of additional timeslots increase. In this example the number of additional timeslots required varies between 0 (node 13) and 510 (node 22) timeslots. This represents a varying overhead of between 0%(node 13) and 10% (node 22). This overhead effectively reduces the radar function time.

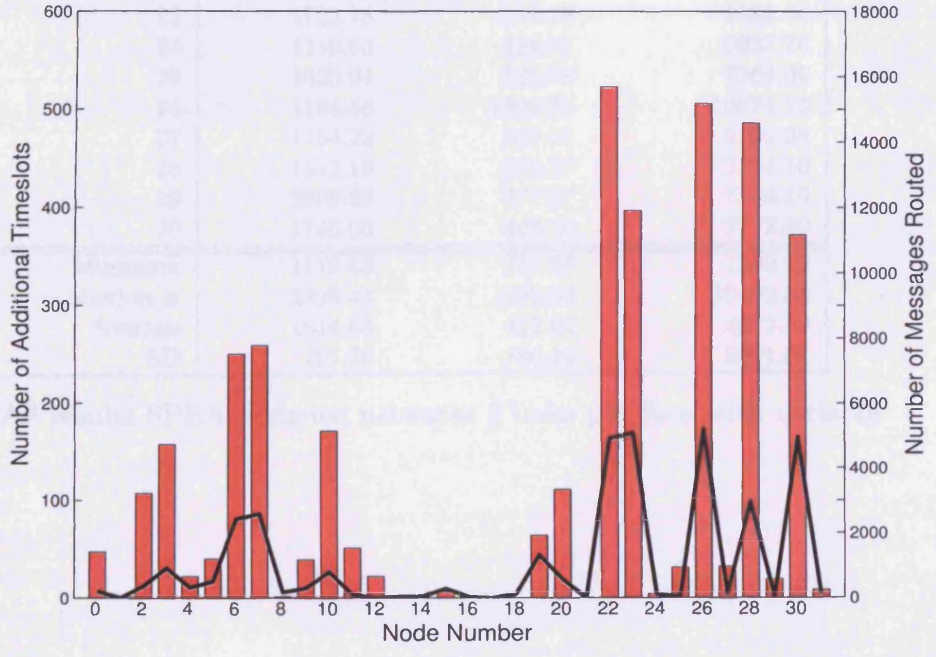



Figure 9.4: Additional timeslots (red bars). Number of messages routed (black lines)

## 9.3 Full SPEA 2 Links Per Face Results

Table 9.3 shows the full results for all thirty test networks. The majority of average delay values fall within the target threshold of 500ms.

| Graph Number | Avg Total No. of extra timeslots used | Avg msg delay With Variable Timeslots (ms) | Avg msg delay Without Variable Timeslots(ms) |
|---|---|---|---|
| 1 | 1882.81 | 352.44 | 5660.33 |
| 2 | 1834.00 | 374.39 | 3082.08 |
| 3 | 1411.88 | 524.95 | 2664.83 |
| 4 | 1742.03 | 374.47 | 5011.44 |
| 5 | 1965.06 | 391.64 | 5763.58 |
| 6 | 2328.44 | 579.89 | 10492.84 |
| 7 | 1770.50 | 334.68 | 5047.65 |
| 8 | 1939.69 | 407.60 | 5948.24 |
| 9 | 1544.53 | 359.59 | 6992.08 |
| 10 | 1502.13 | 363.14 | 2000.17 |
| 11 | 1446.41 | 340.76 | 4241.72 |
| 12 | 1449.81 | 352.91 | 4195.76 |
| 13 | 1429.84 | 334.98 | 6305.85 |
| 14 | 1604.59 | 336.69 | 2796.35 |
| 15 | 1479.66 | 311.55 | 1590.20 |
| 16 | 1342.34 | 324.56 | 2834.27 |
| 17 | 1552.31 | 362.83 | 2381.76 |
| 18 | 1552.31 | 362.83 | 3799.27 |
| 19 | 1186.41 | 340.91 | 4356.71 |
| 20 | 1560.06 | 353.96 | 2082.53 |
| 21 | 1912.91 | 390.26 | 7463.03 |
| 22 | 1582.88 | 406.51 | 5602.26 |
| 23 | 1762.78 | 357.78 | 4662.56 |
| 24 | 1119.63 | 424.81 | 6937.76 |
| 25 | 1620.94 | 528.58 | 7364.06 |
| 26 | 1194.56 | 1309.58 | 10274.19 |
| 27 | 1454.22 | 369.56 | 6199.08 |
| 28 | 1512.19 | 379.37 | 3351.10 |
| 29 | 2009.28 | 377.27 | 7304.10 |
| 30 | 1746.06 | 349.70 | 2772.00 |
| Minimum | 1119.63 | 311.55 | 1590.20 |
| Maximum | 2328.44 | 1309.58 | 10492.84 |
| Average | 1614.68 | 412.61 | 4972.59 |
| SD | 267.56 | 180.12 | 2301.61 |

Table 9.3: All results SPEA designed networks 2 links per face with variable timeslots

The most significant aspect of variable timeslots is that the number of viable command positions has increased significantly. With SPEA designed networks with 2 links per face almost 430 command positions were unusable out of a possible 30x32 = 960 i.e. 44.7%. With variable timeslots all but 2 command positions are now viable for use in radar networks. This improvement can be seen by the increase in the number of command positions with average delays below 500ms (figure 9.6). compared to the same graphs without variable timeslots (figure 9.5).

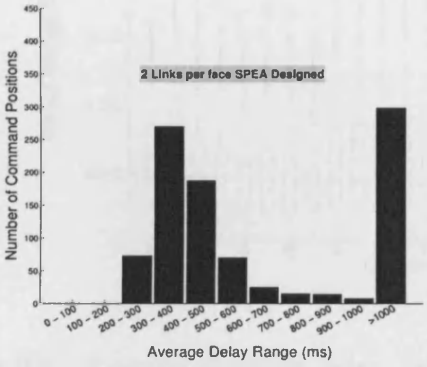



Figure 9.5: SPEA designed graphs with 2 links per face without variable timeslots

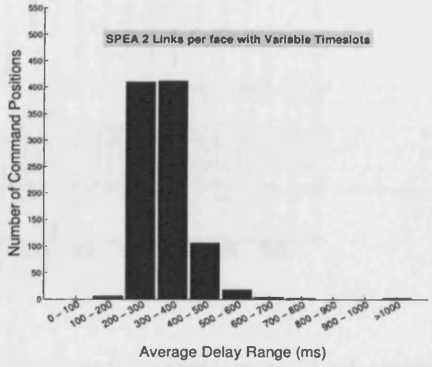



Figure 9.6: SPEA designed graphs with 2 links per face with Variable timeslots

The average delay values of all command positions are also consistent. Figure 9.7 shows the box plot values for all 32 command positions for the thirty SPEA designed test networks with 2 links per face. Figure 9.7 represents the average message delay values of the 960 test commander positions for all graphs. All median and quartile ranges are below 500 ms. The results are typical of variable timeslot performance.

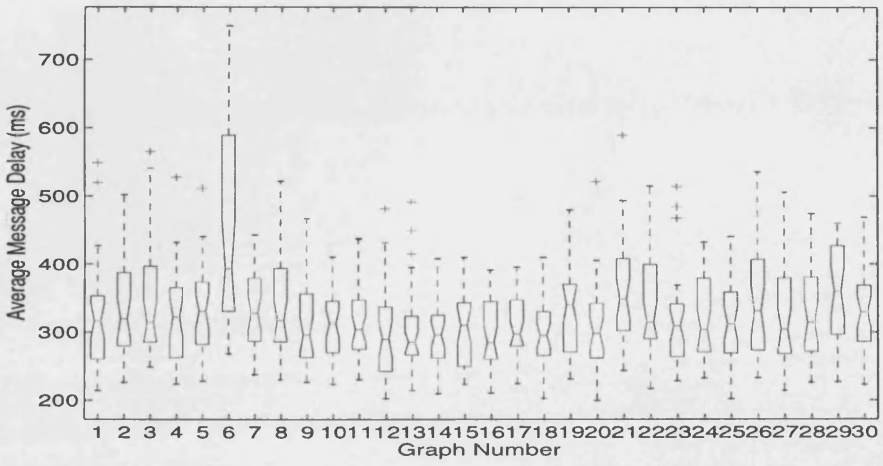



Figure 9.7: Average message delay analysis, all command positions

Figure 9.8 shows the full message analysis of 300000+ messages sent to all 32 command positions for SPEA designed test graph 13 with 2 links per face. Although figure 9.8 does show outline messages that are above 500ms, they represent only 900

out of 300,000 messages or 0.003 percent of all messages. This small percentage is indicative of the quality of service being maintained for the significant majority using variable timeslots.

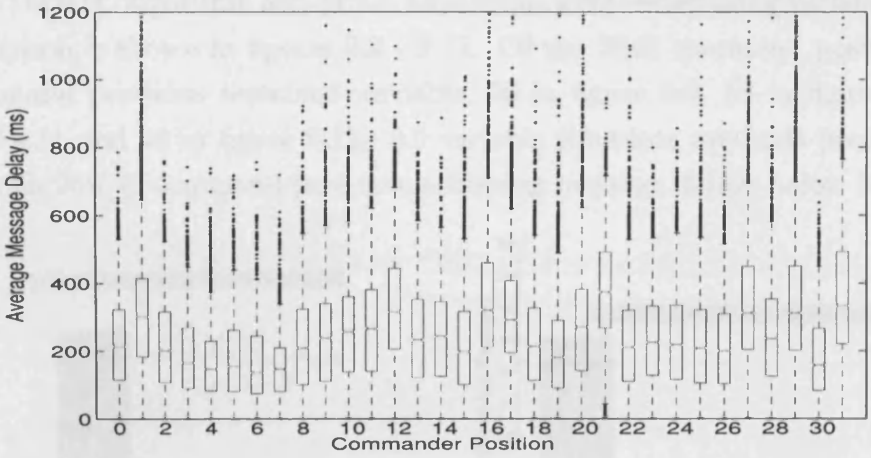



Figure 9.8: Average message delay analysis, all commander positions. Test graph 13 SPEA designed with 2 links per face

## 9.4 Performance Comparison SPEA vs DARTC with Variable Timeslots

Both the DARTC algorithm and SPEA algorithms were tested using variable timeslots. A comparison is shown in figures 9.9 - 9.12. Of the 3840 command positions only 156 command positions remained unviable, 26 in figure 9.9, 58 in figure 9.10, 28 in figure 9.11 and 44 in figure 9.12. All variable timeslots methods produce good results with 96% of command positions achieving message delays below 500ms.

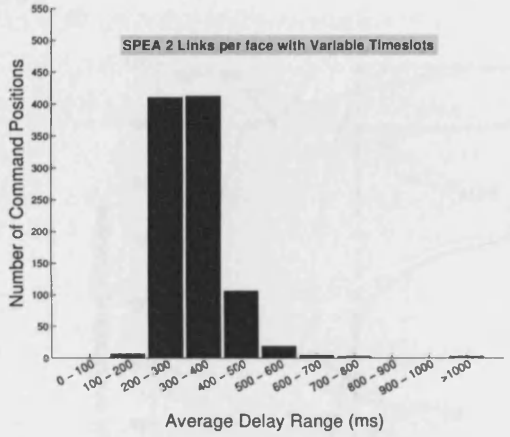



Figure 9.9: SPEA 2LPF with variable timeslots

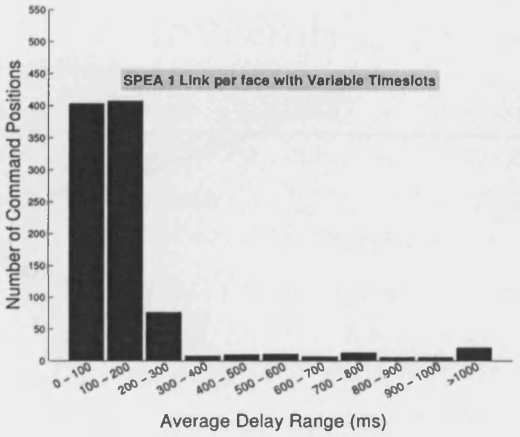



Figure 9.10: SPEA 1LPF with variable timeslots

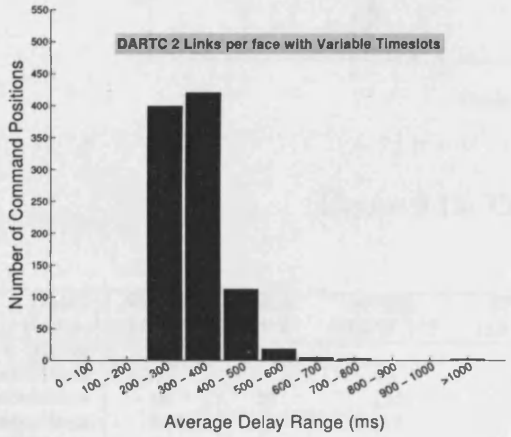



Figure 9.11: DARTC 2LPF with variable timeslots

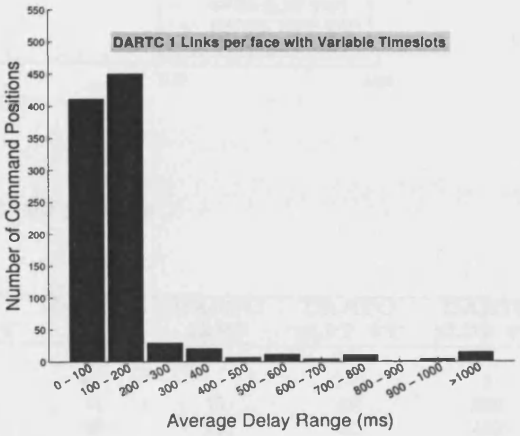



Figure 9.12: DARTC 1LPF with variable timeslots

Figure 9.13 provides a complete comparison of all the topology design methods and their relative performances. Figure 9.13 is the graphical representation of all results for SPEA and DARTC methods with and without variable timeslots. The Table 9.4 shows the delay ranges for all methods. Its is clear from both figure 9.13 and Table 9.4 that 1 link per face has the most number of command positions in the lower average messages delay ranges i.e 0-100ms and 100-200ms. It is also clear that variable timeslots have the highest percentage of viable command positions. This is indicated by the higher percentages at the threshold line (the dashed line in figure 9.13).

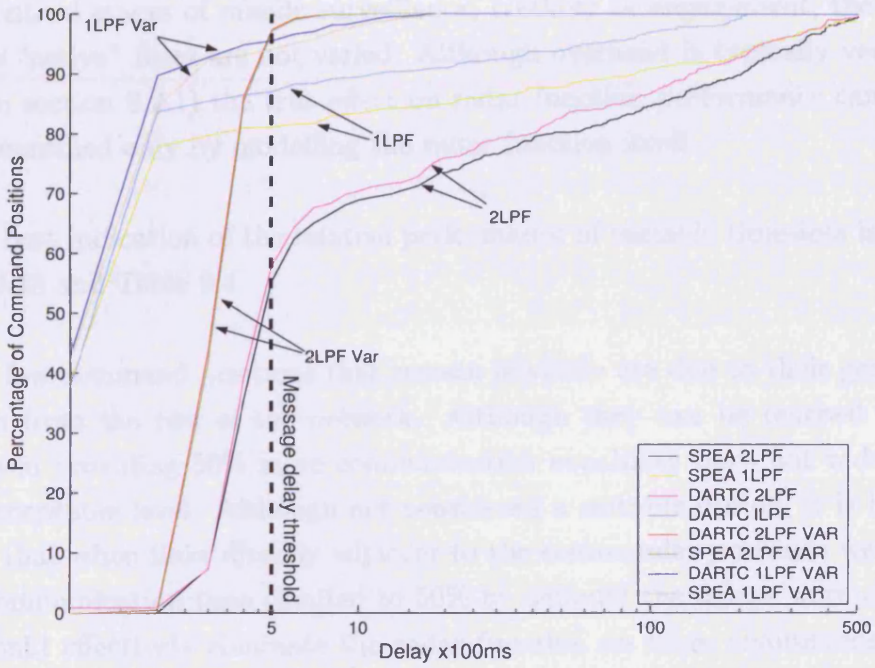



Figure 9.13: Complete results

| Delay Range | SPEA 1LPF | SPEA 2LPF | SPEA 2LPF VT | SPEA 1LPF VT | DARTC 1LPF | DARTC 2LPF | DARTC 1LPF VT | DARTC 2LPF VT |
|---|---|---|---|---|---|---|---|---|
| 0 -100ms | 369 | | | 403 | 362 | | 411 | |
| 100-200ms | 377 | | 6 | 407 | 441 | | 450 | 1 |
| 200-300ms | 18 | 73 | 410 | 76 | 17 | 71 | 29 | 399 |
| 300-400ms | 12 | 270 | 412 | 7 | 10 | 306 | 20 | 420 |
| 400-500ms | 1 | 187 | 106 | 9 | 5 | 177 | 6 | 112 |
| 500-600ms | 5 | 70 | 18 | 10 | 12 | 64 | 11 | 18 |
| 600-700ms | 1 | 25 | 4 | 6 | 1 | 31 | 4 | 5 |
| 700-800ms | 12 | 15 | 2 | 12 | 7 | 7 | 10 | 3 |
| 800-900ms | 3 | 14 | | 5 | 2 | 14 | 1 | |
| 900-1000ms | 1 | 8 | | 5 | 1 | 6 | 4 | |
| >1000ms | 161 | 298 | 2 | 20 | 102 | 284 | 14 | 2 |

Table 9.4: Complete results for SPEA and DARTC with and without variable timeslots

## 9.5 Variable Timeslots Summary

Varying the length of the communication timeslots effectively increases bandwidth. This has the effect of maintaining performance regardless of geographical location of the node (provided they are connected) or commander position. Figures 9.7 and 9.8 show that messages using variable timeslots are consistently below the required 500ms threshold in most cases.

By allowing the timeslots length to be varied only when required, their effect on radar performance is minimised. The variable timeslot method also ensures that during critical stages of missile surveillance, tracking or engagement, the timeslots on these "active" faces are not varied. Although overhead is typically very low (as shown in section 9.2.1) the true effect on radar function performance can be accurately quantified only by modelling the radar function itself.

The best indication of the relative performance of variable timeslots is shown in Figure 9.13 and Table 9.4.

The few command positions that remain unviable are due to their geographical isolation from the rest of the network. Although they can be reached with long links, even providing 50% more communication capability does not reduce delays to an acceptable level. Although not considered a suitable option, it is interesting to note that when links directly adjacent to the commander positions were allowed 100% communication time (limited to 50% by default) the delays were eliminated. This would effectively eliminate the radar function on these communicating faces but would allow messages from the radar network to be received under the threshold 500ms. The effect of carrying out such a drastic measure can be evaluated only by modelling the radar function more precisely.

Therefore, whilst the idea of variable timeslots seems appropriate from a communications perspective, its true effect on radar function can only be ascertained by evaluating the radar function using a more accurate functional radar model. Variable timeslots are able to compensate for the majority of delays and congestion associated with the networks tested. This is also achieved with low total overheads in terms of additional communication time required. It is also clear that only few radar nodes in any network require the majority of the additional timeslots. This results in very few individual radar nodes being significantly affected but as a radar network, the total radar performance improves dramatically. This is highlighted by the fact that 97% of all command positions (using the test networks) achieve average message delays less than 500ms.

CHAPTER

# TEN

# Scalability

This project has focused on networks covering 28 x 28$km$ [80]. The expected number of radars to enable coverage of this area is not expected to be greater than 32 radar nodes[80]. This area is expected to be under the control of a single commander. Therefore design aims are geared towards being able to design networks consisting of 32 radar nodes only. The simulation environment is also geared towards providing accurate analysis of 32 node networks with ability to vary the number and direction of targets. Moderate expansion to extend network coverage can be expected and has been discussed in previous sections. However, increasing network coverage significantly has not been considered necessary. This section looks briefly at some of the issues involved with increasing network size in terms of algorithm design performance and actual network performance. As the network size increases, the coverage increases. The obvious consequence is that the total number of targets tracked by the network increases while the target density on average remains the same [80]. This results in an increase in network traffic routed to the commander and consequently increases link utilisation. It is therefore difficult to compare performance of different network sizes due to the variation in network traffic and link load.

## 10.1 Algorithm Scalability

In order to test algorithm scalability, a series of random node placements were generated for different network sizes. The nodes were placed on flat terrain. Various network sizes were considered up to 256 nodes. Only ten random node placements were generated for node sizes above 128 nodes due to the excessive time required to generate the network designs. The following details the node sizes tested and number of random networks generated:

- 16 nodes - 20 random node placements
- 32 nodes - 20 random node placements
- 64 nodes - 20 random node placements
- 96 nodes - 20 random node placements
- 128 nodes - 20 random node placements
- 160 nodes - 10 random node placements
- 192 nodes - 10 random node placements
- 224 nodes - 10 random node placements
- 256 nodes - 10 random node placements

Both the SPEA and DARTC design methods were then applied to these test places and average design times recorded. No network performance evaluation using the simulator was carried out. The average design time results of the tests are shown below in Table 10.1 and Figures 10.1 and 10.2.

| No. of Nodes | Avg. Time (secs) | |
|---|---|---|
| | SPEA | DARTC |
| 16 | 4 | 0.1 |
| 32 | 28 | 0.27 |
| 64 | 90 | 0.51 |
| 96 | 270 | 0.88 |
| 128 | 726 | 1.25 |
| 160 | 2155 | 1.52 |
| 192 | 4732 | 1.99 |
| 224 | 7955 | 2.55 |
| 256 | 14829 | 3.44 |

Table 10.1: Comparison SPEA vs DARTC average design times (secs)

For this comparison in Table 10.1, the SPEA was run for 1000 generations and the DARTC algorithm was run to completion.

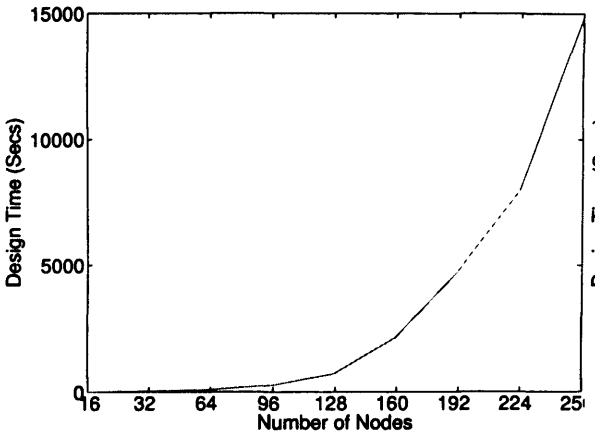



Figure 10.1: SPEA Scalability

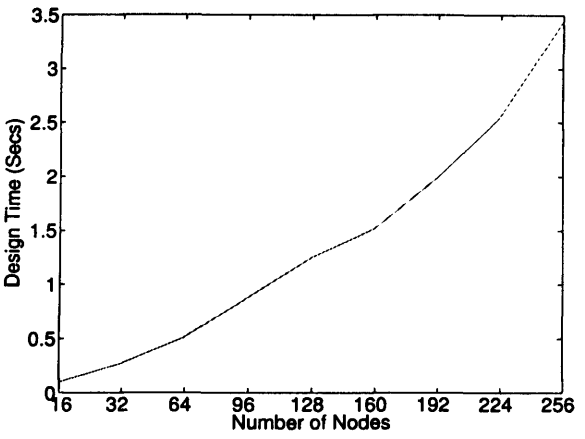



Figure 10.2: DARTC Scalability

The SPEA algorithm has been tailored to networks consisting of 32 node networks. Being able to produce network designs for networks consisting of 32 node networks under 60 seconds is an important aim of this project (as defined in section 1). Increasing network size increases the complexity and because some network evaluation operations are $O(n^2)$ where n is the number of nodes in the design, network design times can increase dramatically. With the increases in network size comes increases in the number of repair operations. The increasing complexity of the network encoding and running of the SPEA algorithm itself also contributes to poorer performance with larger network sizes. The SPEA algorithm performance rapidly deteriorates above 64 nodes as can be seen in both Table 10.1 and Figure 10.1.

The DARTC algorithm in contrast, although design times do not scale linearly, does not increase above an acceptable level. The simplistic nature and operation of the network design algorithm allow good performance for the network sizes tested. Table 10.1 and Figure 10.2 allows even the largest of the networks considered to be designed in under 5 seconds. If fully implemented, the DARTC algorithm could potentially run faster due to the distributed nature of the algorithm. Design times would be limited only by the hardware constraints and time required to pass link setup messages and coordinating setup procedures.

## 10.2 Scalability of Network Performance

From section 10.1 it is clear that a distributed approach would be most appropriate as network sizes increase. Although this project is primarily concerned with network sizes of 32 nodes, larger network sizes are considered here but the node characteristics remain unchanged. It is, however, difficult to make comparisons between performance, under simulation, of 32 nodes and larger networks. This is because, as network sizes increase, the number of expected targets increases and therefore the total amount of traffic increases. The number of messages per second arriving at the commander also increases.

In order to analyse how the increase in network sizes affect simulation performance, a few simple examples are considered. Table 10.2 details the number of networks tested for three different network sizes. As network sizes increase, the simulation times increase dramatically. The network simulator uses an event-driven simulation system and model. Therefore, as the number of targets and nodes increases, the number of events increase. All other networking operations such as messages routing and route updates increase in complexity, and hence time taken to complete the simulations increase. For this reason, the number of networks and commander positions are kept low to ensure simulation completes in a reasonable time. All simulations are carried out on flat terrain.

To increase the simulation speed for large networks, a table driven simulation method would have to be used. This would mean that at a particular time a predefined message is sent from a particular node without the simulation of the radar event. Cahn [9] details methods of traffic modelling based on either a probabilistic method of traffic generation or using actual network traffic and applying it to a particular network design. Both methods result in a table defining traffic flowing between particular nodes at particular times. This is not possible in this thesis as probabilistic methods require good knowledge of the application. As the netted radar concept is relatively new, this application knowledge is not available.

Using existing traffic, which essentially requires running a full simulation and then forming a traffic table, would result in major alterations in how the simulator currently operates and is beyond the scope of these initial investigations.

| No. of Nodes | Area Coverage $km^2$ | No. Networks Tested | No.Command Positions Tested | Total Simulation Time (hours) | No. Targets |
|---|---|---|---|---|---|
| 64 | 1600 | 4 | 256 | 168 | 20 |
| 128 | 3136 | 2 | 128 | 312 | 28 |
| 256 | 6400 | 1 | 10 | 217 | 40 |

Table 10.2: Networks tested for performance scalability

Network command positions are tested with and without variable timeslots. The simulation using variable timeslots requires the entire simulation to be re-run for all command positions effectively doubling the simulation time. Therefore simulation times are effectively double that shown in Table 10.2.

### 10.2.1 64 Node Networks

Figures 10.3 to 10.6 show the four test networks built using the DARTC algorithm with 1 link per face.

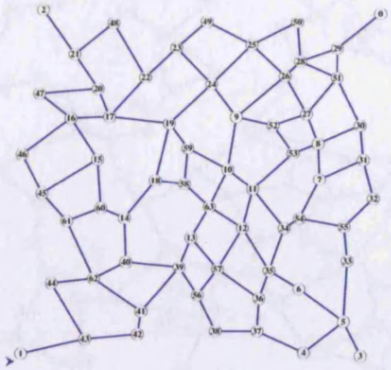


Figure 10.3: 64 Node Test Network 0

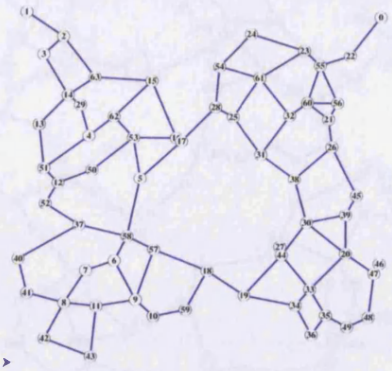


Figure 10.4: 64 Node Test Network 1

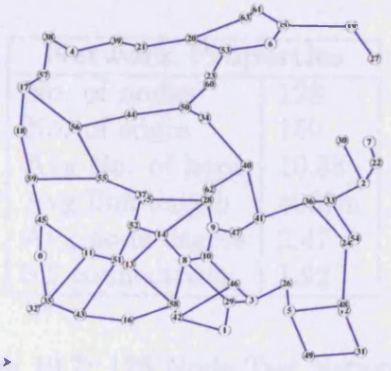


Figure 10.5: 64 Node Test Network 2

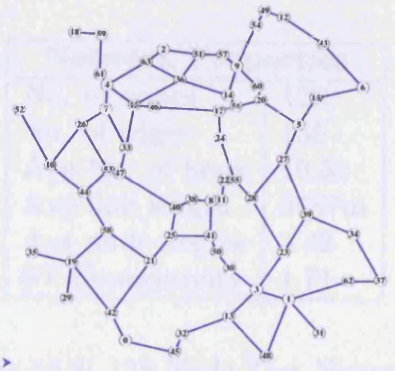


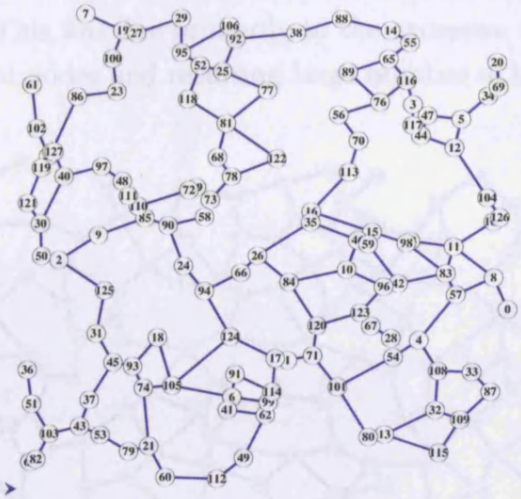
Figure 10.6: 64 Node Test Network 3

Table 10.3 shows the simulation results for the networks above. 78% [calculated $(47 + 55 + 49 + 51)/(4 \times 64)$] of command positions are viable without variable timeslots, 93% [calculated $(60 + 62 + 57 + 58)/(4 \times 64) = 93$] of command positions are viable with variable timeslots. The average number of messages received by the commander increases to over 15000 messages.

| 64 Node Test Network | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| Average link length | 4223.93 | 3953.57 | 4183.35 | 3973.81 |
| Average number of hops | 5.39 | 6.20 | 6.82 | 6.08 |
| All-pairs ST connectivity | 2.40 | 2.02 | 1.94 | 2.04 |
| Radar coverage in surveillance mode | 98.70% | 95.90% | 96.20% | 94.80% |
| Number of messages received by commander | 15334 | 14958 | 15304 | 15176 |
| Number of viable command positions without variable timeslots | 47 | 55 | 49 | 51 |
| Number of viable command positions with variable timeslots | 60 | 62 | 57 | 58 |
| Max. Number of additional timeslots used | 4598 | 5126 | 6369 | 5847 |
| Max. Number of additional timeslots used by single node | 1156 | 1263 | 1328 | 1148 |

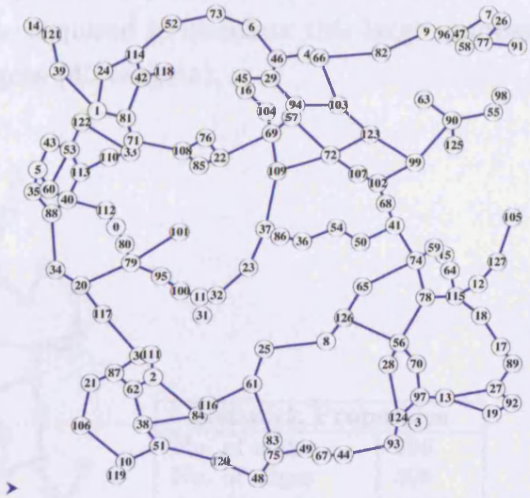
Table 10.3: 64 Node Results

### 10.2.2 128 Node Networks



| Network Properties | |
|---|---|
| No. of nodes | 128 |
| No. of edges | 159 |
| Avg No. of hops | 10.38 |
| Avg link length | 4087m |
| Avg node degree | 2.47 |
| ST connectivity | 1.92 |

Figure 10.7: 128 Node Test Network 0



| Network Properties | |
|---|---|
| No. of nodes | 128 |
| No. of edges | 155 |
| Avg No. of hops | 10.51 |
| Avg link length | 3987m |
| Avg node degree | 2.42 |
| ST Connectivity | 1.71 |

Figure 10.8: 128 Node Test Network 1

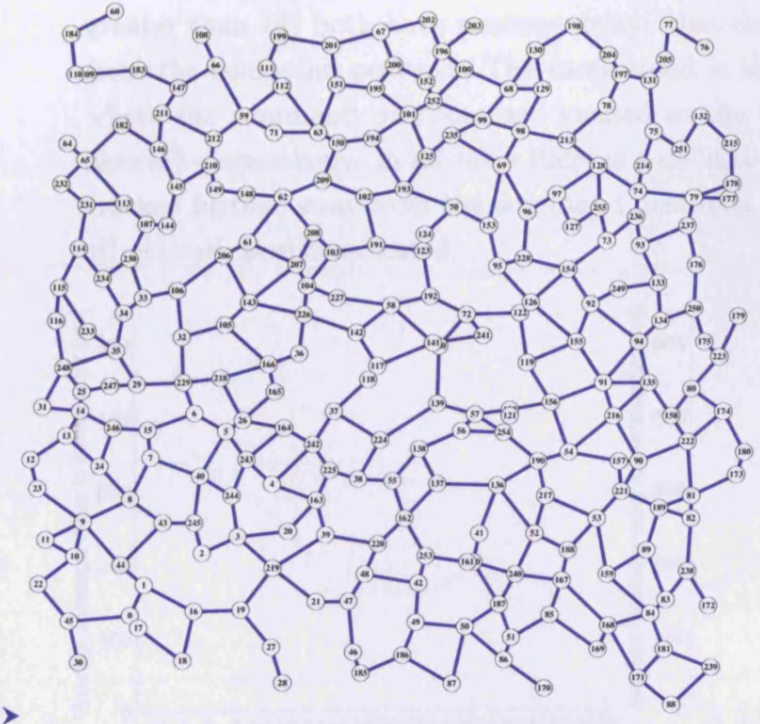
Table 10.4 shows the results for 128 randomly chosen command positions out of a possible 256. Without variable timeslots 60% (calculated $(38+39)/(2\times 64) = 60$) of command positions tested were viable compared to 92%(calculated $(59+58)/(2\times 64) = 92$) with variable timeslots.

| Graph number | 0 | 1 |
|---|---|---|
| Radar coverage in surveillance mode | 97.20 | 98.10 |
| Average number of messages received by commander | 30446 | 31083 |
| Number of command positions tested | 64 | 64 |
| Number of viable command positions without variable timeslots | 38 | 39 |
| Number of viable command positions with variable timeslots | 59 | 58 |
| Max. Number of extra timeslots used | 9079 | 9686 |
| Max. Number of additional timeslots used by single node | 1912 | 2058 |

Table 10.4: 128 Node Results

### 10.2.3 256 Node Networks

Only one 256 node with 10 commander positions picked at random was tested. This was due primarily to the excessive time required to simulate this large number of nodes and resulting large number of targets (40 targets).



| Network Properties | |
|---|---|
| No. of nodes | 256 |
| No. of edges | 398 |
| Avg No. of hops | 10.61 |
| Avg link length | 4417m |
| Avg node degree | 3.03 |
| ST connectivity | 2.84 |

Figure 10.9: 256 Node DARTC design with 98% coverage of $6400km^2$

Table 10.5 shows results for ten command positions. The average number of messages sent to each commander node was approximately 65000, with the message rate peaking at about 230 messages per second. This causes severe congestion along links closest to the commander as they are the most utilised links. Again, by implementing variable timeslots, 8 out of 10 command positions were viable.

| Commander ID | Without Variable Timeslots | With Variable Timeslots | Total no. of additional timeslots |
|---|---|---|---|
| 8 | >30000ms [1] | 452.65 | 10615 |
| 58 | 495.02 | 245.27 | 14919 |
| 75 | >30000ms | 416.45 | 15249 |
| 99 | >30000ms | 336.46 | 16663 |
| 126 | >30000ms | 312.45 | 17176 |
| 167 | >30000ms | 383.79 | 182872 |
| 173 | >30000ms | 1095.34 | 19109 |
| 199 | >30000ms | 1962.42 | 19346 |
| 219 | >30000ms | 363.94 | 17181 |
| 229 | >30000ms | 398.12 | 18322 |

Table 10.5: 256 Node Results

What can be illustrated with this large number of links is how the average message delay of targets varies as they pass over the network. Targets closest to the commander have the best delay with progressively decreasing performance and as the targets move away from the commander the delay increases. Figure 10.10 illustrates that when the commander is situated at node 58, a central location in the network, targets to the North (targets IDs less than 20) and South (targets IDs greater than 24) both have message delays that decrease as the targets are further from the command position. The same trend is shown in Figures 10.11 and 10.12 where the command positions are located in the South and North regions of the network respectively. In all cases there is a definite and gradual trend as targets are tracked further away from the command positions. These results were obtained for all network positions tested.

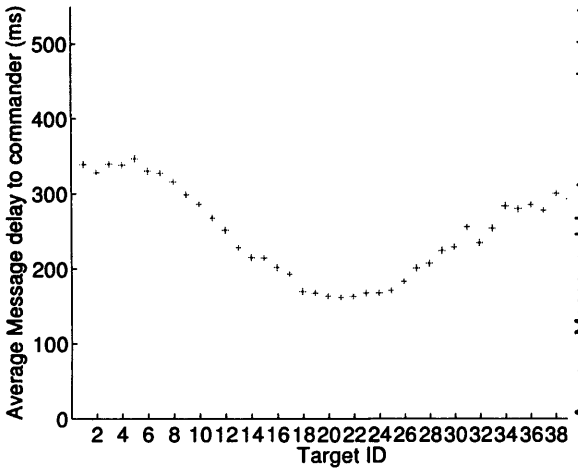



Figure 10.10: Commander at node 58

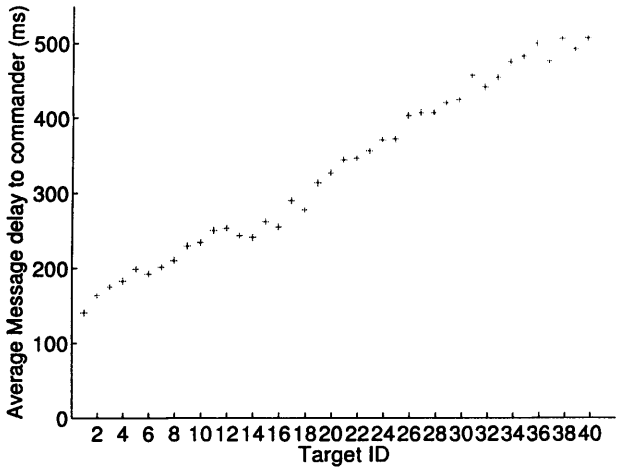



Figure 10.11: Commander at node 167

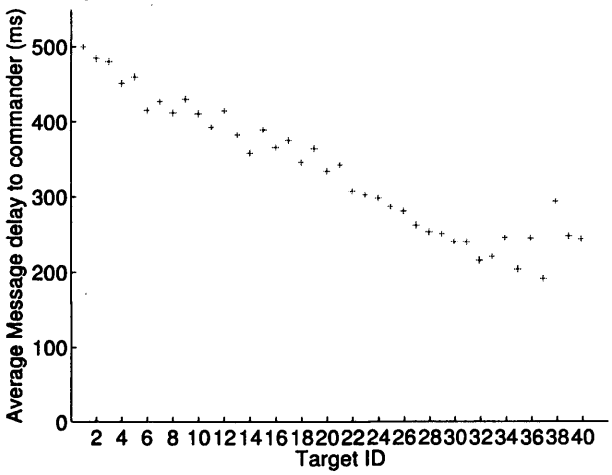



Figure 10.12: Commander at node 99

### 10.2.4 Scalability Summary

Three network sizes were tested consisting of 64, 128 and 256 nodes. All networks were designed using the DARTC algorithm with 1 link per face. All networks had command positions that were viable without variable timeslots. However, with variable timeslots, all network sizes tested achieved comparatively better performance.

From these initial tests it appears that as network sizes increase, the congestion on links adjacent to the commander increases significantly. This renders viable only the best command positions, i.e. the more central, highly connected command positions. With variable timeslots this problem is overcome effectively by increasing bandwidth along these congested links. The variable timeslots method scales well even allowing the largest of the networks (256 nodes) tested to have a high percentage of viable command positions.

CHAPTER

# ELEVEN

## Terrain Effects

Results in previous sections have detailed results for flat terrain. This section looks at the effect terrain has on netted radar design. Terrain is considered only from a line of sight perspective between nodes. Its effects on target detection performance are not evaluated. Evaluating target detection with difficult terrain also involves more complex and accurate target simulation criteria such as target height. When evaluating the targets in three dimensions the terrain obscuration effects would have to be taken into account. For example if a radar node is next to a ridge or hill, the view of the target would be limited by that ridge or hill. Placing a node directly behind such a structure would limit its ariel view. This thesis only considers the targets in two dimensions, if the target is within range of the radar, then it can be detected regardless of where it is in relation to obstructing structures. This reduces the complexity of the simulator.

As more accurate target profiles have not yet been considered, target detection remains solely based on target range. Rather than consider different types of terrain, a simple worst case example is used to determine how both the global and local design methods perform with difficult terrain.

The terrain used is a $784km^2$ section of the Ordnance Survey SH grid. It contains what can be considered very arduous terrain containing high mountain peaks (including Mt. Snowdon), ridges and deep valleys. Together with the large range of heights, the terrain makes lines of sight across the terrain very difficult. Random placement of the nodes in such arduous terrain is unlikely due to the mobility limitations of the radars themselves. Therefore the nodes are placed in a non-random fashion to avoid peaks and to focus placement on regions with gentler slopes.

Test nodes as shown in figure 11.1 are positioned in a non random fashion in order to give good radar coverage and also to avoid regions with a high degree of slope. Figure 11.2 shows a slope analysis of the underlying terrain derived using the GIS. Slope degrees of less than 30 degrees are considered appropriate.

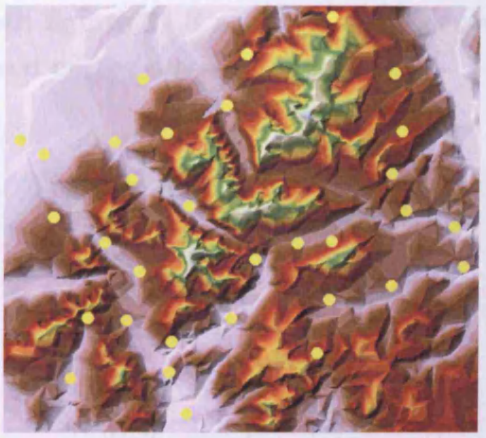


Figure 11.1: Test layout for difficult terrain with underlying surface elevation

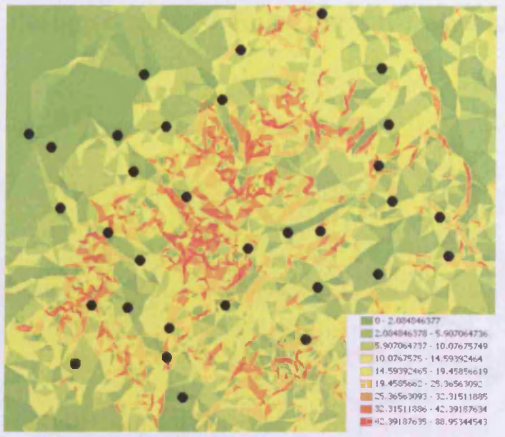


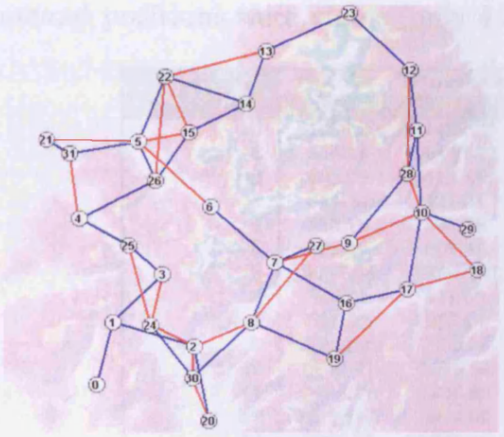
Figure 11.2: Test layout for difficult terrain with terrain slope analysis

In the example terrain given in figure 11.1, it is unlikely that if the nodes where placed in a completely random fashion, they would be able to form a completely connected network. Approximately 50 different random node placements were placed on the terrain shown in figure 11.1 and none was able to form a connected network (a single contiguous network containing all 32 nodes). It is a relatively simple task to manually insert the nodes into flat areas with high visibility, but in a military scenario these might also be the most exposed areas. This conflict of interest and the random node placement issue requires further investigation and background to how these radar nodes would actually be positioned in a military environment.

## 11.1 Radar Network Design with Difficult Terrain

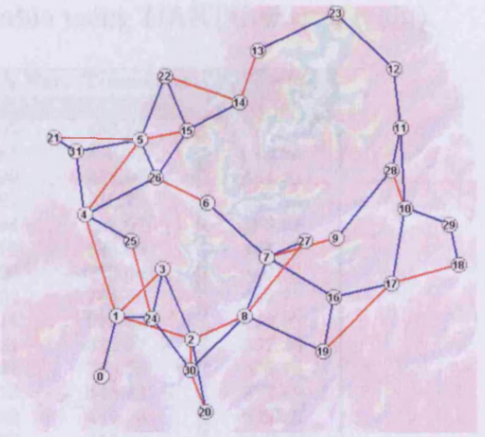
### 11.1.1 Difficult Terrain Example 1

Figure 11.3 shows the network design produced by the local DARTC algorithm. Figure 11.4 shows the network design produced by the global SPEA algorithm. Both figures 11.3 and 11.4 show networks designed with 2 links per face and are based on the terrain in figure 11.1.



| Network Properties | |
|---|---|
| No. of nodes | 32 |
| No. of edges | 53 |
| Avg No. of hops | 3.69 |
| Avg link length | 4641m |
| Avg node degree | 3.3125 |
| ST connectivity | 2.07 |

Figure 11.3: DARTC 2LPF Design



| Network Properties | |
|---|---|
| No. of nodes | 32 |
| No. of edges | 51 |
| Avg No. of hops | 3.76 |
| Avg link length | 4517m |
| Avg node degree | 3.18 |
| ST connectivity | 2.18 |

Figure 11.4: SPEA 2LPF Design

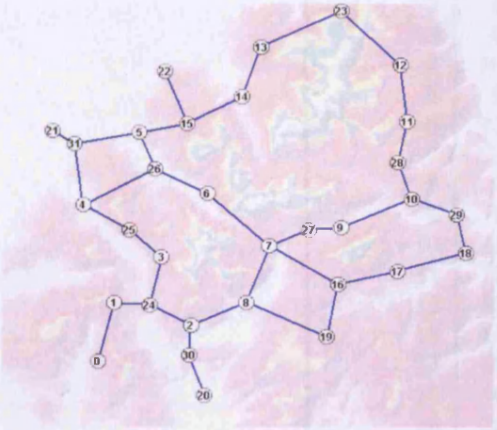
Table 11.1 shows the average message delays to each of the 32 commander positions. Both the SPEA and DARTC algorithms were tested with and without variable timeslots. Without variable timeslots, the SPEA or DARTC algorithms were unable to sustain more than 50% of viable command positions. In contrast, all network positions were viable when variable timeslots are employed.

In this particular simulation the network was designed with no terrain (no line of sight restrictions). The average message delay was significantly lower and 11 command positions were viable (only 4 are viable using DARTC with terrain).

| Commander | No Var. Timeslots | | With Var. Timeslots | | No Terrain |
|---|---|---|---|---|---|
| Position | SPEA | DARTC | SPEA | DARTC | DARTC |
| 0 | 31686.00 | 31469.20 | 378.05 | 480.84 | 22772.70 |
| 1 | 554.29 | 16153.80 | 295.81 | 408.32 | 600.84 |
| 2 | 498.08 | 411.46 | 330.38 | 324.98 | 484.39 |
| 3 | 548.42 | 427.91 | 371.61 | 373.75 | 338.36 |
| 4 | 533.74 | 508.45 | 305.94 | 357.97 | 496.22 |
| 5 | 529.46 | 607.98 | 349.18 | 333.09 | 491.03 |
| 6 | 9567.58 | 31992.50 | 298.20 | 283.04 | 399.73 |
| 7 | 546.56 | 541.60 | 308.64 | 281.86 | 442.71 |
| 8 | 587.72 | 576.14 | 319.11 | 299.26 | 440.96 |
| 9 | 27338.90 | 7063.06 | 411.51 | 372.67 | 557.33 |
| 10 | 2982.37 | 1655.54 | 415.91 | 380.50 | 536.78 |
| 11 | 3168.91 | 1834.80 | 495.76 | 448.14 | 977.49 |
| 12 | 12794.80 | 3924.33 | 493.92 | 449.28 | 3950.20 |
| 13 | 6987.81 | 2287.16 | 453.57 | 458.50 | 2177.38 |
| 14 | 2465.95 | 1232.82 | 413.53 | 457.42 | 1149.98 |
| 15 | 557.61 | 626.69 | 358.56 | 403.67 | 467.97 |
| 16 | 3149.53 | 1810.28 | 348.88 | 342.16 | 830.31 |
| 17 | 3104.89 | 1720.98 | 398.54 | 372.44 | 803.07 |
| 18 | 3155.54 | 22025.70 | 474.57 | 458.22 | 981.37 |
| 19 | 3285.46 | 2385.12 | 355.16 | 358.06 | 2280.14 |
| 20 | 674.31 | 645.90 | 423.52 | 424.02 | 652.54 |
| 21 | 1736.11 | 1782.63 | 453.71 | 488.31 | 666.70 |
| 22 | 656.03 | 707.14 | 391.35 | 351.66 | 425.94 |
| 23 | 32114.60 | 32077.90 | 451.34 | 401.82 | 32030.30 |
| 24 | 461.51 | 379.83 | 337.40 | 319.03 | 367.50 |
| 25 | 3786.19 | 424.45 | 391.13 | 344.45 | 347.73 |
| 26 | 523.15 | 693.31 | 299.65 | 360.67 | 356.65 |
| 27 | 930.38 | 897.14 | 396.42 | 368.21 | 790.02 |
| 28 | 3074.49 | 1765.85 | 438.35 | 389.96 | 735.36 |
| 29 | 3095.37 | 14557.10 | 507.72 | 462.65 | 934.37 |
| 30 | 473.61 | 455.61 | 311.44 | 346.49 | 416.65 |
| 31 | 1672.88 | 1648.48 | 404.51 | 387.76 | 493.85 |
| Minimum | 461.51 | 379.83 | 295.81 | 281.86 | 338.36 |
| Maximum | 32114.60 | 32077.90 | 507.72 | 488.31 | 32030.30 |
| Average | 5101.32 | 5790.34 | 386.98 | 384.04 | 2481.14 |
| SD | 8731.55 | 9889.28 | 62.93 | 56.97 | 6681.99 |

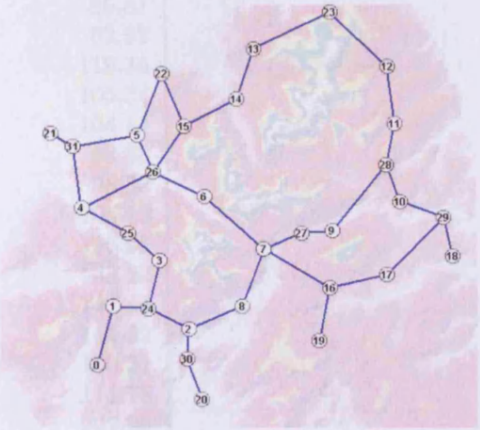
Table 11.1: Average delay and timeslot analysis for all command positions with difficult terrain DARTC 2LPF

Figures 11.5 and 11.6 show network designs for the SPEA and DARTC algorithm using 1 link per face.



| Network Properties | |
|---|---|
| No. of nodes | 32 |
| No. of edges | 36 |
| Avg No. of hops | 4.91 |
| Avg link length | 3881m |
| Avg node degree | 2.25 |
| ST connectivity | 1.61 |

Figure 11.5: DARTC 1LPF design



| Network Properties | |
|---|---|
| No. of nodes | 32 |
| No. of edges | 36 |
| Avg No. of hops | 4.71 |
| Avg link length | 3898m |
| Avg node degree | 2.25 |
| ST connectivity | 1.69 |

Figure 11.6: SPEA 1LPF design

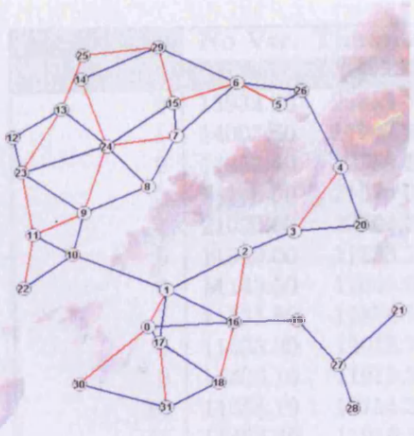
Table 11.2 page 208 shows the average message delay for all command positions. All command positions are viable in this instance and so variable timeslots need not be employed. It should also be noted that the average message delay values are significantly lower than the 2 links per face designs. ST connectivity is lower in both instances than the 2 links per face designs.

| Comm ID | Avg msg delay(ms) SPEA | DARTC |
|---|---|---|
| 0 | 638.60 | 638.00 |
| 1 | 95.16 | 100.65 |
| 2 | 91.23 | 95.81 |
| 3 | 84.20 | 92.92 |
| 4 | 104.77 | 119.35 |
| 5 | 98.85 | 105.34 |
| 6 | 100.07 | 104.14 |
| 7 | 83.84 | 88.88 |
| 8 | 90.43 | 98.33 |
| 9 | 90.74 | 105.19 |
| 10 | 117.73 | 135.65 |
| 11 | 121.52 | 133.32 |
| 12 | 136.04 | 147.44 |
| 13 | 144.16 | 144.32 |
| 14 | 137.59 | 140.31 |
| 15 | 106.73 | 111.78 |
| 16 | 138.12 | 227.96 |
| 17 | 132.88 | 209.48 |
| 18 | 131.07 | 159.27 |
| 19 | 154.32 | 238.60 |
| 20 | 106.69 | 111.61 |
| 21 | 112.33 | 126.71 |
| 22 | 123.93 | 121.02 |
| 23 | 482.53 | 497.11 |
| 24 | 83.57 | 88.41 |
| 25 | 93.68 | 108.78 |
| 26 | 99.05 | 101.43 |
| 27 | 85.75 | 92.63 |
| 28 | 120.78 | 133.88 |
| 29 | 121.04 | 150.89 |
| 30 | 99.92 | 104.98 |
| 31 | 103.24 | 125.25 |
| Minimum | 83.57 | 88.41 |
| Maximum | 638.60 | 638.00 |
| Average | 138.45 | 154.98 |
| SD | 114.16 | 115.90 |

Table 11.2: Average delay for all command positions SPEA vs DARTC 1LPF with difficult terrain

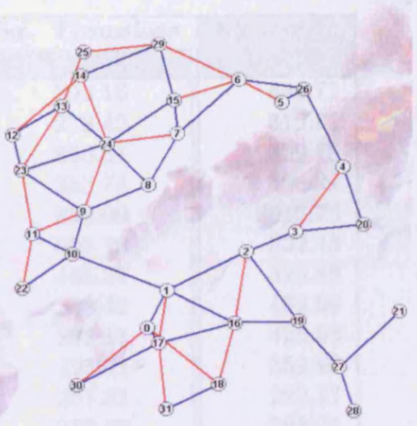
### 11.1.2 Difficult Terrain Example 2

This section uses terrain that has a large ridge which separates the network and regions of fairly flat terrain as can be seen in the overlays in figures 11.7 and 11.8. These figures show network designs for the SPEA and DARTC algorithms using 2 link per face.



| Network Properties | |
|---|---|
| No. of nodes | 32 |
| No. of edges | 53 |
| Avg No. of hops | 3.93 |
| Avg link length | 5305m |
| Avg node degree | 3.31 |
| ST connectivity | 1.95 |

Figure 11.7: DARTC 2LPF design



| Network Properties | |
|---|---|
| No. of nodes | 32 |
| No. of edges | 55 |
| Avg No. of hops | 3.85 |
| Avg link length | 5395m |
| Avg node degree | 3.43 |
| ST connectivity | 1.98 |

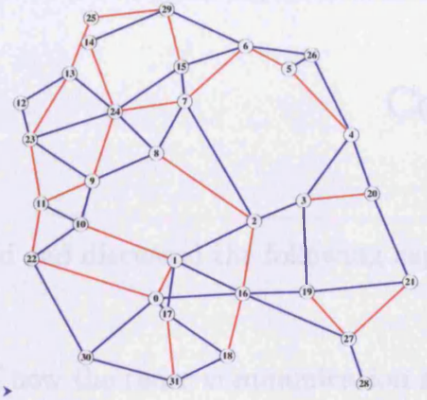
Figure 11.8: SPEA 2LPF design

Table 11.3 shows the average message delays to each of the 32 commander positions. Both the SPEA and DARTC algorithms were tested with and without variable timeslots. Without variable timeslots, neither the SPEA nor DARTC algorithm was able to sustain any viable command positions. In contrast, all network positions apart from commander position 1 were viable when variable timeslots were employed.

| Commander | No Var. Timeslots | | With Var. Timeslots | | No Terrain |
|---|---|---|---|---|---|
| Position | SPEA | DARTC | SPEA | DARTC | DARTC |
| 0 | 13932.90 | 25859.50 | 312.34 | 392.16 | 446.71 |
| 1 | 14003.60 | 17299.20 | 224.74 | 278.42 | 319.81 |
| 2 | 14353.40 | 22659.50 | 295.67 | 865.31 | 359.75 |
| 3 | 18148.00 | 21576.00 | 356.23 | 383.74 | 408.24 |
| 4 | 21020.00 | 22784.20 | 367.59 | 392.00 | 2015.73 |
| 5 | 11359.00 | 11833.10 | 509.03 | 507.79 | 522.43 |
| 6 | 11143.50 | 11641.60 | 439.96 | 445.37 | 426.89 |
| 7 | 11335.80 | 11874.10 | 411.59 | 394.42 | 462.99 |
| 8 | 11433.80 | 12015.70 | 387.00 | 392.11 | 425.63 |
| 9 | 11406.10 | 11919.30 | 302.78 | 325.01 | 359.45 |
| 10 | 11558.10 | 12154.30 | 221.01 | 257.81 | 292.17 |
| 11 | 11483.40 | 11919.50 | 325.19 | 350.52 | 398.01 |
| 12 | 11337.20 | 11864.20 | 451.10 | 471.74 | 517.38 |
| 13 | 11302.80 | 11823.70 | 441.08 | 447.11 | 493.72 |
| 14 | 15671.60 | 31416.60 | 489.64 | 522.19 | 599.93 |
| 15 | 11167.40 | 11643.80 | 437.15 | 429.48 | 412.99 |
| 16 | 13945.80 | 16748.90 | 313.23 | 420.82 | 415.33 |
| 17 | 14046.00 | 22486.30 | 279.21 | 313.21 | 325.59 |
| 18 | 15586.80 | 16964.40 | 395.02 | 401.11 | 435.65 |
| 19 | 25989.40 | 27070.70 | 344.94 | 431.95 | 466.52 |
| 20 | 23283.60 | 24581.50 | 433.15 | 461.84 | 413.35 |
| 21 | 49756.00 | 38432.70 | 506.94 | 490.53 | 11262.16 |
| 22 | 11589.10 | 12136.40 | 366.84 | 413.61 | 430.14 |
| 23 | 11429.90 | 11995.00 | 368.15 | 387.88 | 413.39 |
| 24 | 11365.50 | 11918.30 | 356.70 | 355.57 | 405.07 |
| 25 | 15150.30 | 15368.90 | 505.76 | 505.85 | 520.49 |
| 26 | 11348.10 | 11825.20 | 422.35 | 448.58 | 2652.07 |
| 27 | 32808.00 | 28713.00 | 436.13 | 466.19 | 3552.82 |
| 28 | 32076.50 | 27992.80 | 532.82 | 312.66 | 387.14 |
| 29 | 10958.50 | 11329.60 | 498.08 | 489.70 | 524.34 |
| 30 | 38071.70 | 42158.90 | 442.15 | 491.18 | 10510.75 |
| 31 | 19618.00 | 17380.70 | 441.50 | 439.60 | 474.18 |
| Minimum | 10958.50 | 11329.60 | 221.01 | 257.81 | 292.17 |
| Maximum | 49756.00 | 42158.90 | 532.82 | 865.31 | 11262.16 |
| Average | 17114.99 | 18668.36 | 394.22 | 427.67 | 1301.59 |
| SD | 9278.07 | 8468.09 | 82.54 | 104.61 | 2614.97 |

Table 11.3: Difficult terrain example 2 variable timeslots summary 2LPF (all times in ms)

The last column in Table 11.3 shows the performance of a network designed without the terrain restricted lines of sight. The resulting network design is shown below in figure 11.9. The number of edges spanning what was previously difficult terrain, allow good message passing and reduced message delays. This is shown in Table 11.3 with the drastic improvement in performance and number of viable command positions without using variable timeslots.



| Network Properties | |
|---|---|
| No. of nodes | 32 |
| No. of edges | 60 |
| Avg No. of hops | 3.22 |
| Avg link length | 5762m |
| Avg node degree | 3.85 |
| ST connectivity | 2.26 |

Figure 11.9: DARTC 2LPF design with no terrain affected lines-of-sight

### 11.1.3 Terrain Implications

Primarily, making terrain more difficult (hilly) has the effect of reducing lines of sight. In difficult terrain this can lead to a reduction in suitable radar locations. Difficult terrain such as in section 11.1.1 requires the radars to be placed to give good lines of sight. Whilst this might be an appropriate method to ensure radar coverage, militarily these sites might be the most exposed and vulnerable areas. Random placement of nodes on difficult terrain such as in section 11.1.1 would not be possible.

Although the difficult terrain in the examples described have resulted in a significant drop in performance, more detailed analysis of different areas and types would be beneficial to more accurately assess the effect of terrain.

CHAPTER

# TWELVE

## Concluding Remarks

This thesis has analysed and discussed the following aspects of tightly controlled netted radars:

1. A complete model of how the radar communication subsystem might operate.
2. How inter radar node communication links might be set-up by an automatic network design process.
3. Performance issues that arise with different design methods.
4. Design procedures allowing for node movement and additions.
5. Possible modifications to allow better netted radar communication performance.

The model of netted radar communication subsystem was established based on the original concepts of the basic operation of netted radars. The ideas were more formally specified from a communications perspective and developed into the model used in this thesis. The model developed provides a benchmark model for further development of the netted radar communication concept.

## 12.1 The Model

The model was built based on the original netted radar sensing concept [52] and is considered an accurate representation of how the communication system of a network of radars, consisting of four low powered phased arrays, might operate. By modelling each entity as an object, real world behavior could be more closely simulated. As such, all objects including radar systems, faces, links, messages, targets, terrain, etc were included in the model. This level of complexity allowed detailed simulation analysis of the operation of a netted radar. The development of a custom built simulation environment was also the only feasible option available to simulate such networks. No "off-the-shelf" simulator could support the complexity of the

faces, timeslots or target tracking. Simulating the radar function allowed accurate traffic simulations. By generating traffic using an event driven model, each target could be simulated individually. This also allowed each target direction and velocity to be controlled. Radar nodes could then react to the target as it entered its range. Using this approach allowed message generation and therefore traffic distribution to be accurately modelled.

Discrete event driven models are vital for allowing the nodes to be repositioned and correct traffic to be generated. The consequence of this complexity is that it requires significant computational power. For example, full message analysis of 30 test networks used throughout this thesis takes approximately 2 weeks. Another method could have been to derive a table driven model whereby each network goes through an initial discrete process to define when a message would be sent from a particular node. This would eliminate the constant simulation of the radar nodes and target trajectories. The resulting table of messages could be used to improve simulation speed using a form of look-up table. In retrospect this would seem the appropriate method. However this conclusion could only be made after the experience gained throughout this project. The simulation results remain accurate, only the processing and hence time taken could have been reduced.

The full functionality of the targets has not been utilised since, in order to make any comparisons between networks, it was necessary to maintain a predefined target sequence. Random target generation would not allow the networks to be compared as the traffic density and traffic generation would fluctuate drastically.

The structure of the simulator and model allow different aspects of both the environment and the network to be included. Firstly, the environment is defined, which in this project, is limited to terrain and therefore simulates lines of sight only. Thereafter the nodes are positioned within this environment from which a communications network is built. Targets are then passed over the network and the message processing simulated and analysed. The simulation/evaluation system can be considered a set of input parameters, the terrain, node positions, etc linked together to form the complete system. Results of running the system are recorded as output parameters, for example, which nodes are viable. This complete system is shown in Figure 12.1.

All aspects are flexible, e.g. terrain specific to the deployment area could be used along with the required node positions. The network design could then be built, tested and the command position viability could be analysed. It is therefore conceivable that the entire simulation system could be used as a field based evaluation

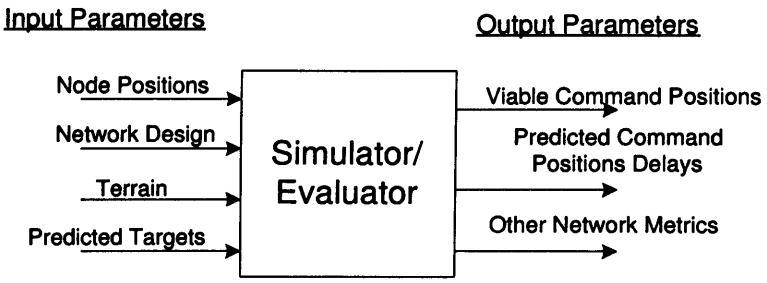



Figure 12.1: Simulation/Evaluation System

tool. It could be used by commanders to decide on the most appropriate place to be situated within the network. With the expected processing power increase in the next decade, the network evaluation times could be reduced to an acceptable level.

In conclusion, the netted radar model, design and subsequent simulation have allowed the research to progress by looking at different aspects of the radar system without having to rely on probabilistic network metrics and procedures. The model therefore behaves in a reactive way and allows all aspects of the system to be varied and controlled.

## 12.2 Netted Radar Design

This thesis has focused on two design methods, a global design method which utilises the SPEA algorithm to form candidate networks, and a local design method (DARTC) that allows nodes to make local decisions to collectively form a radar network. The question as to which method is better cannot simply be answered based on the relative performance of each method. In fact, as can be seen from Figure 9.13, the DARTC method actually outperforms the SPEA method by only 1-5% in terms of number of viable commander positions. In terms of connectivity, SPEA generally outperforms the DARTC method as the global algorithm is able to better control the design sequence from a global perspective.

From a military perspective it is clear that having a central control site for the design process, as with SPEA, immediately renders that site as a prime target for attack. Disabling this site would not allow further designs to be made. This can be overcome by introducing failover sites also capable of radar network design but as it is expected that this capability requires additional computer power compared to normal nodes, this could prove costly.

Initially, global design was thought to be the way forward in order to gain tighter

control over the design process. Further inspection and development of the DARTC algorithm shows that although there is no global control, utilising basic geometric properties and characteristics of the nodes allow well constructed, connected and well performing networks. The DARTC algorithm is also very simple and requires minimal computational power.

The majority of networks tested were based on 32 nodes which is predicted to be the typical network size of this type. If, however, network sizes increase beyond 32 nodes, it would become very difficult to use anything other than a distributed approach such as the DARTC method. The combinatorial explosion which takes place when designing larger networks would render global design infeasible. Section 10 gives a detailed account of how the DARTC algorithm performance scales almost linearly with node size.

Netted radar design is different from traditional network design mainly because of the unknown commander position. As the commander position is varied the messages are routed differently. This prevents the possibility of optimising for a particular command position. The DARTC design algorithm has demonstrated that the best way of designing radar networks for message performance is, in general, to choose the highest capacity links possible. The limited number of faces imposes a further restriction on network design compared to traditional network design. This limitation reduces the possible connections but, as shown in both the SPEA and DARTC radar network design algorithms, both are able to cater for this requirement successfully.

### 12.2.1 Netted Radar Variable Timeslots

Throughout this thesis the difficult command positions have been discussed. These command positions incur large delays due to a number of reasons such as being an edge node with few low capacity links. Congestion along low quality, highly utilised links causes network performance to degrade and rapidly become unacceptable. A modification in the form of variable timeslots (section 9) has improved the simulation performance to the extent that 98% of command positions are viable.

Both the SPEA and DARTC algorithms with 2 links per face when used in conjunction with variable timeslots allow 98% of viable command positions. This improvement is achieved with a less than 1% increase in total communication overhead. By limiting the operation of variable timeslots to nodes that are not tracking targets, their overall impact is kept to a minimum.

### 12.2.2 Increasing Connectivity

Designing high quality and high connectivity radar networks has been a design goal of this thesis. The military nature of the system has meant that reliability becomes a more important factor in forming a radar network. It is obvious from this thesis that designing radar networks with 2 links per face has significantly better connectivity. The disadvantage with 2 links per face is that the networks suffered from a larger number of unviable command positions. In an attempt to overcome connectivity issues with 1 link per face radar networks, a method whereby backup or failover links was suggested. Although the method appears feasible and makes significant improvements in connectivity, even beyond that of 2 links per face radar networks, its impact on actual radar performance needs to be evaluated.

## 12.3 Future work

The model used throughout this project has used a discrete event generation system to simulate the radar function based on the proximity of targets alone. In order to evaluate the performance of the radar function a more comprehensive radar model would have to be employed. This radar model would be based on the evaluation of a target based on the electromagnetic properties of the radar. The target evaluation would then take into account properties such as the orientation of the target (to the radar node) and its radar cross section (RCS). This model would allow the abilities of the netted radar concept to be evaluated not only in terms of message passing delays but also in terms of target detection, tracking, engagement and track quality.

A full radar model would also allow the full evaluation of the variable timeslots method described in this project. If the quality of the radar function could be evaluated then the effect on radar function quality could be established. This analysis would also be applicable to the backup route design method described earlier in this thesis. Backup route design also requires additional radar time and hence a reduction in available radar function time. Therefore, producing a more realistic representation of the radar function model is important in establishing the true quality of the designed netted radar networks and the concept as a whole.

Throughout this thesis the netted radar network model has been tackled from both a local and global perspective. The next stage in deciding which method would be more appropriate would entail a more detailed analysis of exactly how a network might be set-up. This involves issues which are not governed solely by performance of the radar networks but requirements of military operation, hardware capabilities, network control requirements. By deciding at an early stage which method to use,

detailed procedures could be drawn up for how the networks operate in either a global or local design mode. It has been suggested (section 7.2.10) that there might be a hybrid approach using both local design procedures for speed/efficiency and a global method for improving quality. Again by evaluating the global versus local methods in more detail, this type of method could be analysed.

This thesis has assumed that the network links are set-up instantaneously on request. In reality there would be a handshaking procedure to synchronise and correctly establish the links for future communication. Again this might be important because the overheads required to establish links could affect the radar function or communication. Establishing this link setup procedure would then contribute towards a better representation of radar communication function.

Computation complexity of the simulation and evaluation of radar networks has limited the number of simulation and evaluations that can be carried out in a reasonable time. Although being able to dynamically change the properties of the targets is useful, the simulation of this process is very computationally intense. By keeping the target numbers and paths identical for all command positions, messages are generated by the same nodes regardless of the command position (for any given network configuration). Building a message table would allow messages to be generated by looking up entries in this table at any time $t$. Whilst this involves major changes to the operation of the simulator, its benefits could potentially allow much larger numbers of simulations to be completed.

Routing was discussed early in the thesis as a vital element in allowing correct and efficient message delivery. As this netted radar concept is in its infancy, it was thought appropriate to assume instant delivery of routing information. In order to evaluate the impact of the routing information on radar network performance, a routing algorithm needs to be fully implemented. The required routing protocol should be lightweight and able to deal with dynamic networks. This would minimise the overhead required to maintaining effective routing [76].

Terrain has obvious impacts on line of sight restricted applications such as netted radar networks. Few terrain types have been evaluated and therefore, in order to further assess the impact terrain has on the effectiveness of radar networks, more terrain types need to be evaluated. The line of sight also needs to be evaluated from a propagation perspective. The simulator has been designed to allow the inclusion of a propagation model.

— Node additions have been discussed to extend coverage in the form of multiple node additions and singe node additions at arbitrary points in the network. Again the computational complexity limited this evaluation. A more comprehensive treatment of node additions could analyse multiple node additions at arbitrary points. The analysis so far has been limited to effect on message delay only. With a more accurate radar model the effect on radar function itself could be established.

In summary, future work that would be beneficial to establishing the viability of the netted radar concept is outlined below:

1. Form a more realistic radar model.
2. Build a table driven simulation and evaluation system
3. Include a fully implemented routing algorithm.
4. Include a propagation model for the predicted deployment area of the radar network.
5. Determine the effect of additional elements such as variable timeslots and backup routes on radar function.

APPENDIX

# A

# Appendix

## A.1 SPEA and DARTC Test Networks

This section shows the complete set of 30 test network node configurations, SPEA designed networks and DARTC designed networks. All network designs use 2 links per face. Full geographical node positions are available on request.

**Network 1**

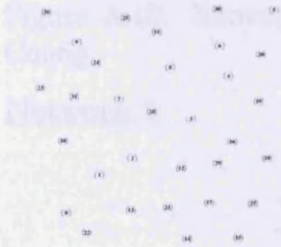


Figure A.1: Network 1 Config

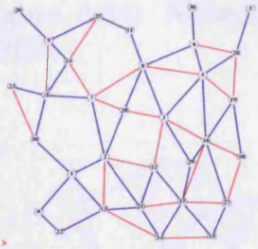


Figure A.2: Network 1 SPEA

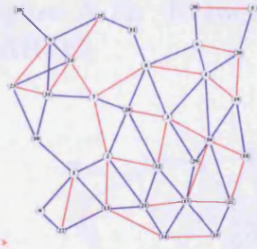


Figure A.3: Network 1 DARTC

**Network 2**

Figure A.4: Network 2 Config

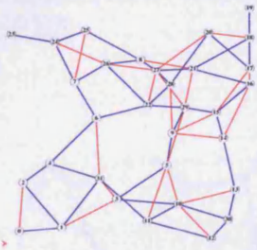


Figure A.5: Network 2 SPEA

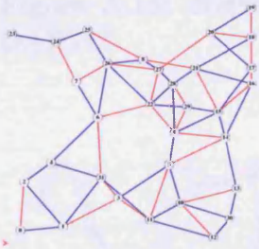


Figure A.6: Network 2 DARTC

**Network 3**

Figure A.7: Network 3 Config

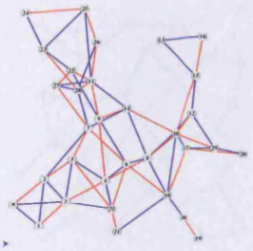


Figure A.8: Network 3 SPEA

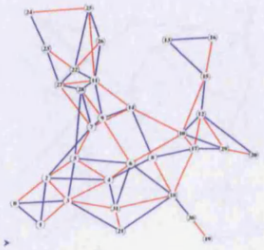


Figure A.9: Network 3 DARTC

**Network 4**

Figure A.10: Network 4 Config

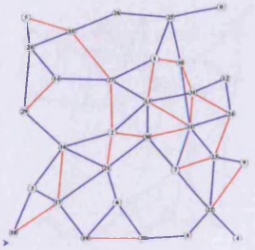


Figure A.11: Network 4 SPEA

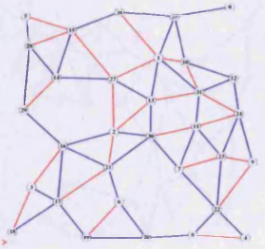


Figure A.12: Network 4 DARTC

**Network 5**

Figure A.13: Network 5 Config

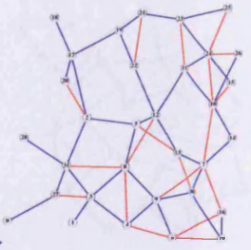


Figure A.14: Network 5 SPEA

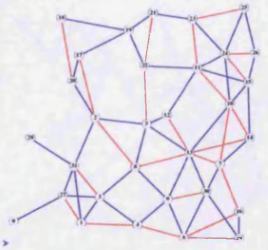


Figure A.15: Network 5 DARTC

**Network 6**

Figure A.16: Network 6 Config

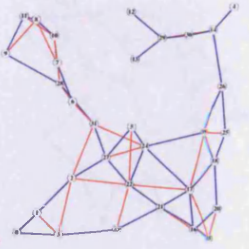


Figure A.17: Network 6 SPEA

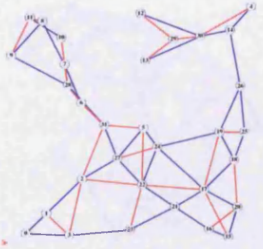


Figure A.18: Network 6 DARTC

**Network 7**

Figure A.19: Network 7 Config

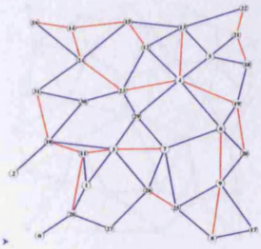


Figure A.20: Network 7 SPEA

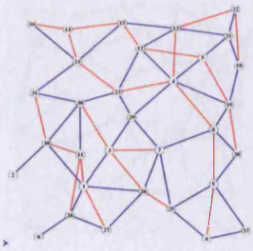


Figure A.21: Network 7 DARTC

**Network 8**

Figure A.22: Network 8 Config

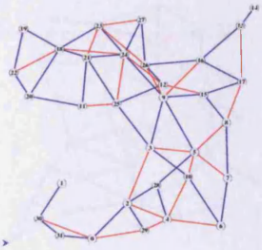


Figure A.23: Network 8 SPEA

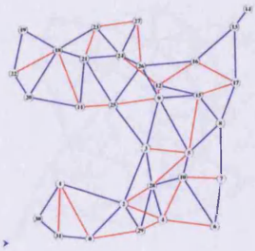


Figure A.24: Network 8 DARTC

**Network 9**

Figure A.25: Network 9 Config

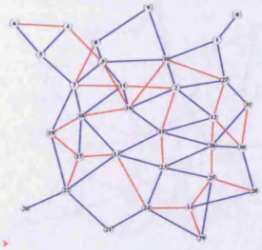


Figure A.26: Network 9 SPEA

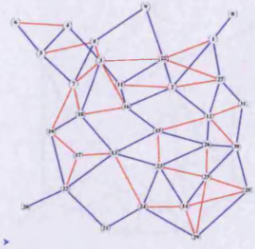


Figure A.27: Network 9 DARTC

**Network 10**

Figure A.28: Network 10 Config

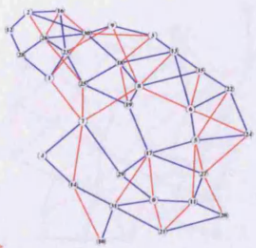


Figure A.29: Network 10 SPEA

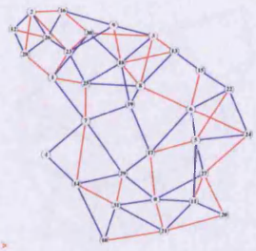


Figure A.30: Network 10 DARTC

**Network 11**

Figure A.31: Network 11 Config

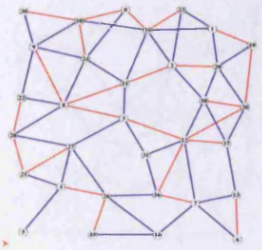


Figure A.32: Network 11 SPEA

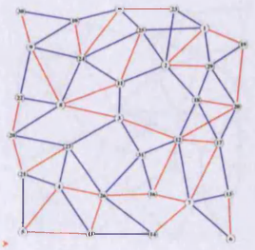


Figure A.33: Network 11 DARTC

**Network 12**

Figure A.34: Network 12 Config

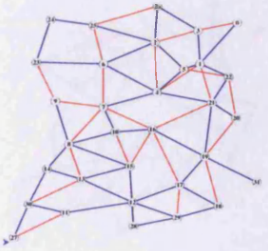


Figure A.35: Network 12 SPEA

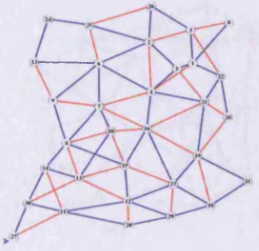


Figure A.36: Network 12 DARTC

**Network 13**

Figure A.37: Network 13 Config

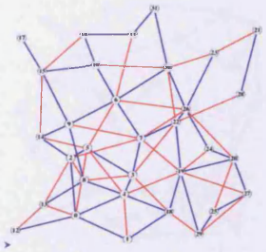


Figure A.38: Network 13 SPEA

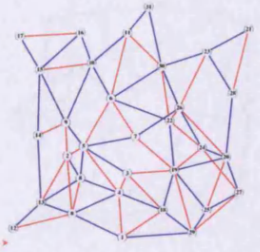


Figure A.39: Network 13 DARTC

**Network 14**

Figure A.40: Network 14 Config

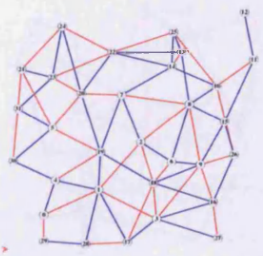


Figure A.41: Network 14 SPEA

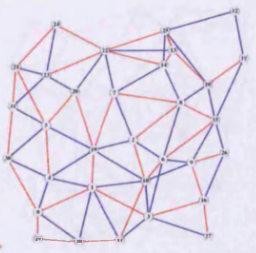


Figure A.42: Network 14 DARTC

**Network 15**

Figure A.43: Network 15 Config

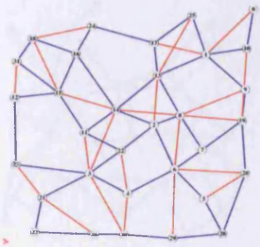


Figure A.44: Network 15 SPEA

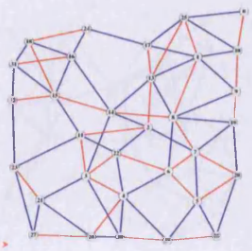


Figure A.45: Network 15 DARTC

**Network 16**

Figure A.46: Network 16 Config

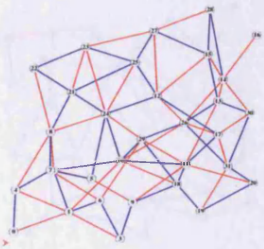


Figure A.47: Network 16 SPEA

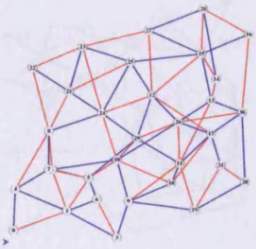


Figure A.48: Network 16 DARTC

**Network 17**

Figure A.49: Network 17 Config

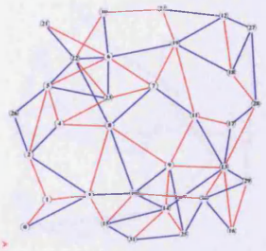


Figure A.50: Network 17 SPEA

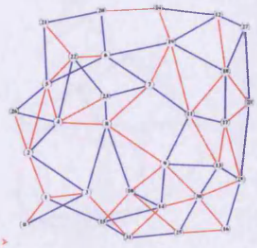


Figure A.51: Network 17 DARTC

**Network 18**

Figure A.52: Network 18 Config

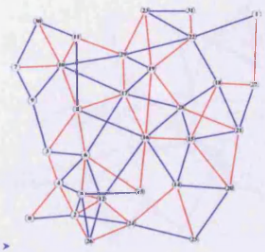


Figure A.53: Network 18 SPEA

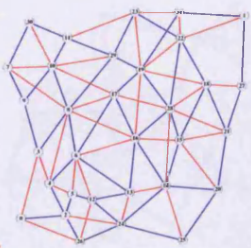


Figure A.54: Network 18 DARTC

**Network 19**

Figure A.55: Network 19 Config

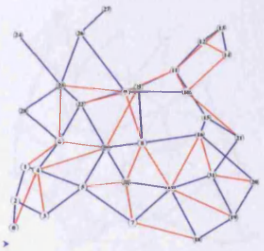


Figure A.56: Network 19 SPEA

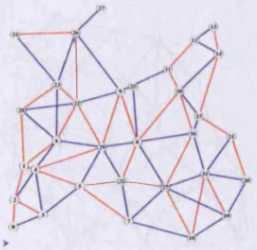


Figure A.57: Network 19 DARTC

**Network 20**

Figure A.58: Network 20 Config

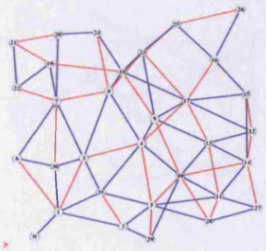


Figure A.59: Network 20 SPEA

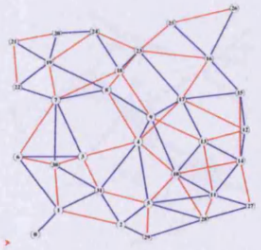


Figure A.60: Network 20 DARTC

**Network 21**

Figure A.61: Network 21 Config

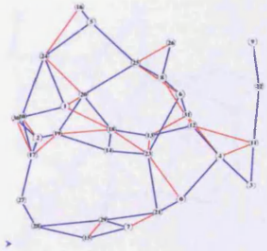


Figure A.62: Network 21 SPEA

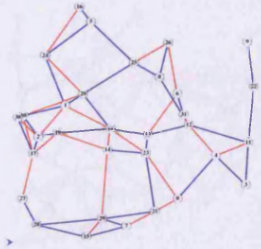


Figure A.63: Network 21 DARTC

**Network 22**

Figure A.64: Network 22 Config

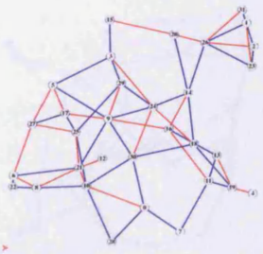


Figure A.65: Network 22 SPEA

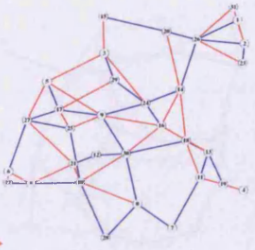


Figure A.66: Network 22 DARTC

**Network 23**

Figure A.67: Network 23 Config

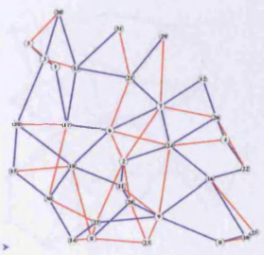


Figure A.68: Network 23 SPEA

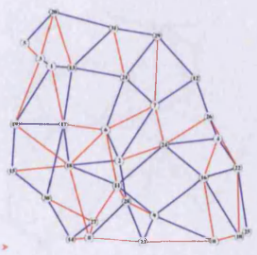


Figure A.69: Network 23 DARTC

**Network 24**

Figure A.70: Network 24 Config

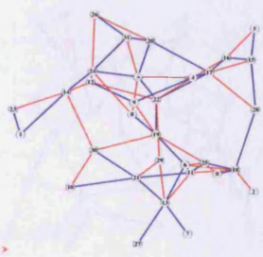


Figure A.71: Network 24 SPEA

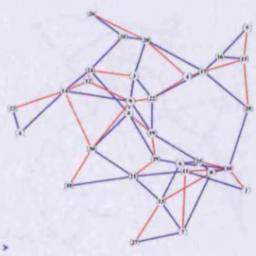


Figure A.72: Network 24 DARTC

**Network 25**

Figure A.73: Network 25 Config

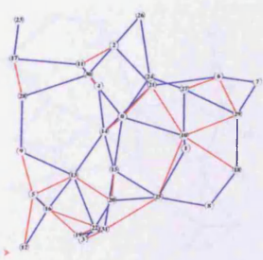


Figure A.74: Network 25 SPEA

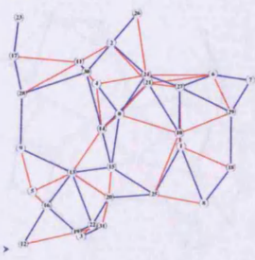


Figure A.75: Network 25 DARTC

**Network 26**

Figure A.76: Network 26 Config

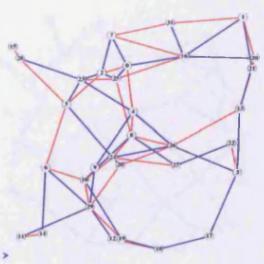


Figure A.77: Network 26 SPEA

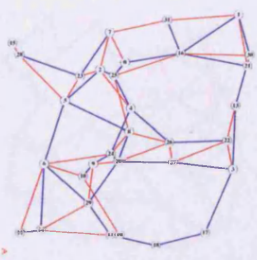


Figure A.78: Network 26 DARTC

**Network 27**

Figure A.79: Network 27 Config

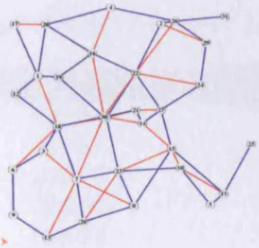


Figure A.80: Network 27 SPEA

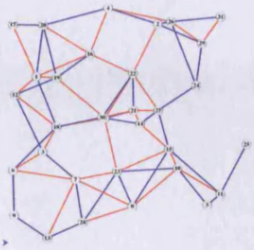


Figure A.81: Network 27 DARTC

**Network 28**

Figure A.82: Network 28 Config

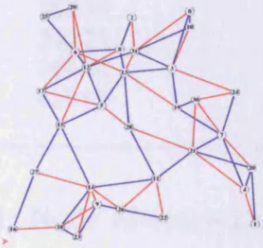


Figure A.83: Network 28 SPEA

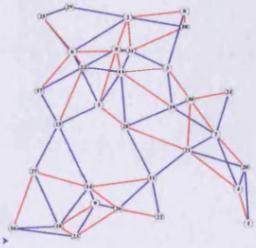


Figure A.84: Network 28 DARTC

**Network 29**

Figure A.85: Network 29 Config

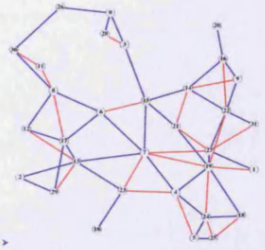


Figure A.86: Network 29 SPEA

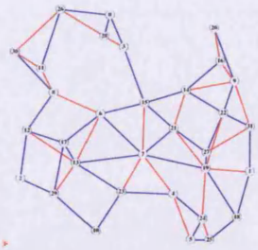


Figure A.87: Network 29 DARTC

**Network 30**

Figure A.88: Network 30 Config

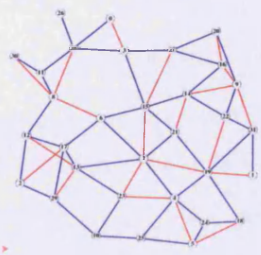


Figure A.89: Network 30 SPEA

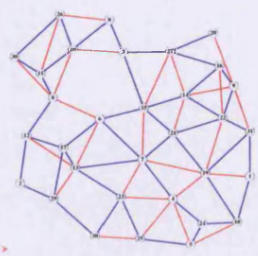


Figure A.90: Network 30 DARTC

## A.2 Worked Results Node Delays

The following results are message delays for each node

| Node Num | Msgs Sent | Avg Delay (ms) | Node Num | Msgs Sent | Avg Delay (ms) |
|---|---|---|---|---|---|
| 0 | 245 | 607.16 | 16 | 239 | 380.89 |
| 1 | 245 | 527.81 | 17 | 256 | 351.96 |
| 2 | 225 | 503.95 | 18 | 225 | 396.56 |
| 4 | 259 | 380.00 | 19 | 226 | 374.19 |
| 5 | 200 | 390.94 | 20 | 260 | 510.84 |
| 6 | 228 | 387.29 | 21 | 260 | 256.28 |
| 7 | 246 | 575.09 | 22 | 189 | 473.76 |
| 8 | 247 | 349.91 | 23 | 182 | 480.81 |
| 9 | 246 | 790.88 | 24 | 239 | 625.21 |
| 10 | 229 | 688.47 | 25 | 221 | 544.11 |
| 11 | 257 | 401.12 | 26 | 168 | 672.63 |
| 12 | 259 | 167.55 | 27 | 259 | 421.62 |
| 13 | 255 | 363.79 | 28 | 259 | 314.66 |
| 14 | 168 | 359.46 | 29 | 252 | 334.80 |
| 15 | 170 | 425.53 | 30 | 192 | 395.02 |
| | | | 31 | 256 | 842.91 |

Table A.1: Node Message Delay Values

## A.3 Standard Deviation

The Table below shows the commander positions and average message delay for test network 13. The standard deviation values are very high due to the large range in average message delay at different command positions.

| Commander Position | Average Message Delay (ms) |
|---:|---:|
| 0 | 28937.70 |
| 1 | 383.14 |
| 2 | 294.36 |
| 3 | 269.69 |
| 4 | 399.87 |
| 5 | 43348.10 |
| 6 | 468.15 |
| 7 | 365.82 |
| 8 | 351.58 |
| 9 | 526.44 |
| 10 | 3532.14 |
| 11 | 825.02 |
| 12 | 344.69 |
| 13 | 407.30 |
| 14 | 449.63 |
| 15 | 498.93 |
| 16 | 311.21 |
| 17 | 320.03 |
| 18 | 773.68 |
| 19 | 426.63 |
| 20 | 501.61 |
| 21 | 378.84 |
| 22 | 18884.50 |
| 23 | 9354.53 |
| 24 | 396.37 |
| 25 | 862.17 |
| 26 | 38524.60 |
| 27 | 436.31 |
| 28 | 307.06 |
| 29 | 358.87 |
| 30 | 27515.50 |
| 31 | 376.05 |
| Average | 5660.328 |
| Standard Deviation | 11906.24 |

## A.4 Full DARTC Results

Full DARTC results with 2 links per face

| Graph Number Number | Average Link Length | Average No. of hops | All pairs ST Connectivity | Average Message delay (ms) | Standard Deviation Avg Msg Delay |
|---|---|---|---|---|---|
| 1 | 5115.28 | 3.19 | 2.60 | 3843.82 | 8988.03 |
| 2 | 4832.93 | 3.19 | 2.87 | 2740.26 | 6185.29 |
| 3 | 4346.49 | 3.11 | 2.31 | 4562.97 | 7416.82 |
| 4 | 5007.94 | 3.30 | 2.54 | 3453.47 | 9447.68 |
| 5 | 5036.96 | 3.14 | 2.77 | 4925.89 | 12860.28 |
| 6 | 4694.84 | 4.22 | 1.61 | 10340.09 | 14139.54 |
| 7 | 5246.67 | 3.19 | 2.56 | 4571.80 | 10492.18 |
| 8 | 4787.97 | 3.42 | 2.33 | 5270.92 | 9297.17 |
| 9 | 4907.83 | 3.09 | 2.93 | 5338.83 | 10652.73 |
| 10 | 4579.03 | 3.07 | 3.45 | 1614.30 | 4234.30 |
| 11 | 5018.33 | 3.06 | 2.91 | 2193.85 | 5301.22 |
| 12 | 4914.47 | 3.02 | 2.99 | 2565.15 | 5918.71 |
| 13 | 4704.56 | 2.94 | 3.11 | 3869.46 | 7629.49 |
| 14 | 4783.60 | 2.90 | 3.42 | 2464.62 | 6466.90 |
| 15 | 4782.57 | 2.91 | 3.03 | 1419.30 | 5459.03 |
| 16 | 4691.07 | 2.75 | 3.61 | 1782.87 | 4255.33 |
| 17 | 4805.04 | 2.90 | 3.64 | 1353.98 | 4314.87 |
| 18 | 4774.85 | 2.73 | 3.39 | 2907.00 | 7723.66 |
| 19 | 4483.43 | 3.12 | 2.68 | 3777.21 | 10850.84 |
| 20 | 4803.96 | 2.85 | 3.36 | 2076.89 | 7007.91 |
| 21 | 4288.07 | 3.41 | 2.46 | 8038.50 | 13478.07 |
| 22 | 4441.98 | 3.30 | 2.22 | 5743.19 | 7335.94 |
| 23 | 4675.86 | 2.94 | 2.89 | 3376.76 | 9422.18 |
| 24 | 4384.44 | 3.05 | 2.53 | 5953.29 | 10551.93 |
| 25 | 4525.75 | 3.07 | 2.85 | 5697.31 | 9221.83 |
| 26 | 4376.74 | 3.19 | 2.76 | 6019.69 | 10679.72 |
| 27 | 4670.23 | 3.07 | 2.63 | 4224.86 | 10046.75 |
| 28 | 4644.48 | 3.07 | 2.46 | 4236.68 | 6966.05 |
| 29 | 4699.91 | 3.23 | 2.55 | 4416.35 | 7436.94 |
| 30 | 4992.30 | 3.02 | 2.79 | 1749.24 | 5177.64 |
| Minimum | 4288.07 | 2.73 | 1.61 | 1353.98 | |
| Maximum | 5246.67 | 4.22 | 3.64 | 10340.09 | |
| Average | 4733.92 | 3.12 | 2.81 | 4017.62 | |
| SD | 238.51 | 0.27 | 0.45 | 2032.25 | |

Table A.2: DARTC - 30 networks tested on flat terrain. All networks are designed with 2 links per face

Full DARTC results with 1 links per face

| Graph Number Number | Average Link Length | Average No. of hops | All pairs ST Connectivity | Average Message delay (ms) | Standard Deviation Avg Msg Delay |
|---|---|---|---|---|---|
| 1 | 4600.40 | 3.97 | 1.72 | 634.94 | 2039.08 |
| 2 | 4135.40 | 4.47 | 1.43 | 448.01 | 981.67 |
| 3 | 3534.84 | 4.81 | 1.50 | 1193.24 | 4765.55 |
| 4 | 4739.18 | 4.05 | 1.40 | 1140.36 | 4402.12 |
| 5 | 4493.84 | 4.15 | 1.67 | 1622.31 | 5870.94 |
| 6 | 3836.90 | 6.53 | 1.08 | 5902.11 | 9025.41 |
| 7 | 4635.98 | 4.11 | 1.93 | 475.17 | 2052.60 |
| 8 | 4196.75 | 4.52 | 1.51 | 1734.58 | 4030.13 |
| 9 | 4266.12 | 3.99 | 1.60 | 1189.23 | 5231.22 |
| 10 | 3947.02 | 4.17 | 2.02 | 142.16 | 241.79 |
| 11 | 4425.21 | 4.02 | 2.02 | 973.49 | 4433.22 |
| 12 | 4317.75 | 4.19 | 1.62 | 123.40 | 32.36 |
| 13 | 4067.53 | 3.78 | 1.86 | 408.27 | 1789.56 |
| 14 | 4054.86 | 4.07 | 1.88 | 382.65 | 1314.37 |
| 15 | 4166.27 | 3.95 | 1.39 | 95.14 | 22.23 |
| 16 | 3778.62 | 3.96 | 1.16 | 89.29 | 49.27 |
| 17 | 4131.48 | 3.99 | 1.69 | 101.10 | 15.23 |
| 18 | 4101.87 | 4.08 | 2.16 | 1477.80 | 7105.79 |
| 19 | 3887.11 | 4.30 | 1.64 | 2606.80 | 7896.57 |
| 20 | 4203.88 | 3.93 | 1.74 | 941.80 | 4725.17 |
| 21 | 3741.62 | 4.82 | 1.57 | 2753.90 | 8294.82 |
| 22 | 3707.21 | 5.00 | 1.43 | 2537.69 | 4823.42 |
| 23 | 3988.04 | 4.11 | 1.92 | 1700.60 | 6264.80 |
| 24 | 3511.13 | 5.71 | 1.36 | 3864.57 | 9300.05 |
| 25 | 3834.24 | 4.60 | 1.52 | 3278.28 | 8103.97 |
| 26 | 3476.33 | 5.08 | 1.11 | 4062.63 | 9810.80 |
| 27 | 4075.01 | 4.19 | 1.62 | 1683.05 | 5983.62 |
| 28 | 3805.24 | 4.69 | 1.43 | 1146.82 | 2461.45 |
| 29 | 4199.47 | 4.51 | 1.68 | 3940.34 | 10261.16 |
| 30 | 4467.94 | 3.97 | 1.71 | 118.01 | 38.25 |
| Minimum | 3476.33 | 3.78 | 1.08 | 89.29 | |
| Maximum | 4739.18 | 6.53 | 2.16 | 5902.11 | |
| Average | 4077.57 | 4.39 | 1.61 | 1558.92 | |
| SD | 331.76 | 0.59 | 0.27 | 1468.25 | |

Table A.3: DARTC - 30 networks tested on flat terrain. All networks are designed with 2 links per face

# BIBLIOGRAPHY


[1] R. M. Pathak A. Kumar and Y. P. Gupta. Genetic algorithm based reliability optimization for computer network expansion. *IEEE Transactions on Reliability*, 44:63–72, 1995.

[2] Y. P. Gupta A. Kumar, R. M. Pathak and H. R. Parsaei. A genetic algorithm for distributed system topology design. *Computers and Industrial Engineering*, 28:659–670, 1995.

[3] Edoardo Amaldi, Antonio Capone, and Federico Malucelliz. Optimizing UMTS radio coverage via base station configuration. *PIMRC02*, 2002.

[4] Colburn Ball and Provan. Network reliability. *Handbook of Operations Research and Management Science*, 7(11), 1992.

[5] M.O. Ball. Complexity of network reliability computations. *Networks*, 10:153–165, 1980.

[6] William G Bath. Tradoffs in radar networking. *Radar Conferance*, 2002.

[7] Bolt Beranek and Newman's. Open shortest path first. *Request For CommentsFC 1247*, 1988.

[8] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Mobile Computing and Networking*, pages 85–97, 1998.

[9] Robert Cahn. *Wide Area Network Design.* Morgan Kaufmann Publishers inc, 1998.

[10] E. Amaldi A. Capone and F. Malucelli. Planning umts base station location: Optimization models with power control and algorithms. *IEEE Transactions on Wireless Communications*, Vol.2:pages 939–952, 2003.

[11] Jae-Hwan Chang and Leandros Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *INFOCOM (1)*, pages 22–31, 2000.

[12] Xiuzhen Cheng and Ding-Zhu Du. Virtual backbone-based routing in multihop ad hoc wireless networks. citeseer.nj.nec.com/504757.html.

[13] King Tim KitSang Cheung-Yau. Using genetic algorithms to design mesh networks. *IEEE Transactions on networking*, 30:56–61, 1997.

[14] V Chvatal. Tough graphs and hamiltonian circuits. *Discrete Math*, 54:215–228, 1973.

[15] Jared Cohon. Multiobjective programming and planning. *Mathematics in Science and Engineering Academic Press*, 140, 1978.

[16] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms.* MIT Electrical and Computer Science Series. MIT Press, 1992. ISBN 0-262-03141-8.

[17] David W. Corne, Joshua D. Knowles, and Martin J. Oates. The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, J. J. Merelo, and Hans-Paul Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 839–848, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.

[18] Intel Corp. Deploying wlans one size does not fit all http://www.intel.com/ebusiness/pdf/it/wp032001.pdf, 2003.

[19] C.R.Reeves. *Modern Heuristic Techniques for Combinatorial Problems.* McGraw Hill, 1995.

[20] L. Davis. *Handbook of Genetic Algorithms.* van Nostrand Reinhold New York, 1991.

[21] Kalyanmoy Deb. Evolutionary Algorithms for Multi-Criterion Optimization in Engineering Design. In Kaisa Miettinen, Marko M. Mäkelä, Pekka Neittaanmäki, and Jacques Periaux, editors, *Evolutionary Algorithms in Engineering and Computer Science*, pages 135–161. John Wiley & Sons, Ltd, Chichester, UK, 1999.

[22] B. Dengiz, F. Altiparmak, and A. E. Smith. Local search genetic algorithm for optimal design of reliable networks. *IEEE Trans. on Evolutionary Computation*, 1(3):179–188, 1997.

[23] Reinhard Diestel. *Graduate Texts in Mathematics: Graph Theroy.* MIT Press, Springer, 2000.

[24] In D.Z. Du and F. Hwang. Voronoi diagrams and delaunay triangulations. *Computing in Euclidean Geometry*, (1):pages 193–234, 1992.

[25] D. Dubhashi, D. Grable, and A. Panconesi. Near optimal distributed edge colouring via the nibble method, 1995. Theoretical Computer Science Special ESA 95 Issue, 203:225 - 251, 1998.

[26] Lothar Thiele Eckart Zitzler. An evolutionary algorithm for multiobjective optimization: The strength pareto approach. Technical Report 43, Gloriastrasse 35 CH-8092 Zurich, Switzerland, 1998.

[27] Lothar Thiele Eckart Zitzler. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

[28] David W. Corne (Editor) Martin J. Oates (Editor) George D. Smith (Editor). *Telecommunications Optimization: Heuristic and Adaptive Techniques.* Wiley, 2000.

[29] A. Eisenblatter and A. M. C. A. Koster. Fap web - a website devoted to frequency assignment. http://fap.zib.de/.

[30] L. R. Esau and K. C. Williams. On teleprocessing system design part II: a method for approximating the optimal network. *IBM Systems Journal*, 5(3):142–147, 1966.

[31] Kevin Fal. Network simulators.
*Web document*, 2002. http://www.cs.berkeley.edu/ kfall/netsims.html.

[32] Carlos M. Fonseca and Peter J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Genetic Algorithms: Proceedings of the Fifth International Conference*, pages 416–423. Morgan Kaufmann, 1993.

[33] Carlos M. Fonseca and Peter J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1):1–16, 1995.

[34] Carlos M. Fonseca and Peter J. Fleming. Multiobjective optimization and multiple constraint handling with evolutionary algorithms—Part I: A unified formulation. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 28(1):26–37, 1998.

[35] Stephanie Forrest and Melanie Mitchell. What makes a problem hard for a genetic algorithm? some anomalous results and their explanation. *Machine Learning*, 13:285–319, 1993.

[36] Nicholas Fourikis. *Phased Array-Based Systems and Applications.* Wiley Interscience, 1997.

[37] Michael J. Ryan Michael R. Frater. *Tactical Communications for the Digitized Battlefield.* Artech House, 2002.

[38] Kershenbaum A Kermani P Grover GA. Mentor: an algorithm for mesh network topological optimization and routing. *IEEE Transactions on Communications*, 39:503–513, 1991.

[39] Brahim Ghribi and Luigi Logrippo. Understanding GPRS: the GSM packet radio service. *Computer Networks (Amsterdam, Netherlands: 1999)*, 34(5):763–779, 2000.

[40] F Glover. Tabu search. *ORSA Journal on Computing*, 1:190–206, 1989.

[41] CDMA Development Group. http://www.cdg.org/.

[42] IEEE 802.11 Working Group.
wireless ethernet. http://grouper.ieee.org/groups/802/11/.

[43] M. Grtschel, C.L. Monma, and M. Stoer. Design of survivable networks. *Handbooks in Operations Research and Management Science.* citeseer.nj.nec.com/441028.html.

[44] Z. Haas. A new routing protocol for the reconfigurable wireless networks, 1997. In Proc. of the IEEE Int. Conf. on Universal Personal Communications., Oct 1997.

[45] Z. Haas, J. Deng, B. Liang, P. Papadimitatos, and S. Sajama. Wireless ad hoc networks, 2002. In John Proakis, editor, Encyclopedia of Telecommunications John Wiley, December 2002.

[46] Michael Pilegaard Hansen. Tabu Search in Multiobjective Optimisation : MOTS. In *Proceedings of the 13th International Conference on Multiple Criteria Decision Making (MCDM'97)*, Cape Town, South Africa, 1997.

[47] C. Hedrick. Routing information protocol. *STD 34, RFC 1058 ,Rutgers University*, June 1988.

[48] J Holland. Adpatation in natural and artificial systems. *University of Michigan Press*, 1975.

[49] J Horn. Multicriteria decision making and evolutionary computation. *In Handbook of Evolutionary Computation. Institute of Physics Publishing*, 1997.

[50] Jeffrey Horn, Nicholas Nafpliotis, and David E. Goldberg. A Niched Pareto Genetic Algorithm for Multiobjective Optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, volume 1, pages 82–87, Piscataway, New Jersey, 1994. IEEE Service Center.

[51] Zhuochuan Huang, Chien-Chung Shen, Chavalit Srisathapornphat, and Chaiporn Jaikaeo. Topology control for ad hoc networks with directional antennas. 11th Int. Conf on Computer Communications and Networks ICCCN 2002.

[52] A. L. Hume and C. J. Baker DERA. Netted radar sensing. In *Poster1*, pages 416–423. IEEE Radar Conference, 2001.

[53] IEEE. Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Computer Society LAN MAN Standards Committee*, (1), 1997.

[54] V. Jacobson. Compressing tcp/ip headers for low-speed serial links. *Internet Request for Comments*, (1145), 1990.

[55] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[56] D Karger. A randomized fully polynomial time approximation scheme for the all terminal network reliability problem. *SIAM review*, 43(3):179–188, 2001.

[57] Aaron Kershenbaum. *Telecommunications Network Design Algorithms.* McGraw Hill, 1993.

[58] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science, Number 4598, 13 May 1983*, 220, 4598:671–680, 1983.

[59] L Kleinrock. *Queueing Systems Vol II Computer Applications.* John Wiley and Sons (New York), 1976.

[60] Joshua Knowles and David Corne. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In Peter J. Angeline, Zbyszek Michalewicz, Marc Schoenauer, Xin Yao, and Ali Zalzala, editors, *Proceedings of the Congress on Evolutionary Computation*, volume 1, pages 98–105, Mayflower Hotel, Washington D.C., USA, 6-9 1999. IEEE Press.

[61] J B Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7:48–50, 1956.

[62] Li Li, Joseph Y. Halpern, Paramvir Bahl, Yi-Min Wang, and Roger Wattenhofer. Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks. In *Proceedings of the twentieth annual ACM symposium on Principles of distributed computing*, pages 264–273. ACM Press, 2001.

[63] Pei Lofquist Lin and Erickson. Network topology management for mobile ad hoc networks with directional links. In *Milicom Procedings*, pages 85–97, 2002.

[64] R. Mathar and T. Niessen. Optimum positioning of base stations for cellular radio networks. *Wireless Networks vol 6 no 6 pp 421-428*, 2000.

[65] Z. Michalewicz. *Genetic Algorithms Data structures = Evolutionary Programs.* Springer-Verlag, 1996.

[66] Melenie Mitchell. *An introduction to Genetic Algorithms.* MIT Press, 1996.

[67] M. Mouly and M.B. Pautet. The GSM System for Mobile Communication. *Cell And Systems ISBN: 2-9507190-0-7 Bay Foreign Language Books*, 1992.

[68] K. Mehlhorn S. Naher. *LEDA A platform for combinatorial and geometric computing.* Cambridge University Press, 2000.

[69] D. Eppstein Z. Galil G. F. Italiano A. Nissenzweig. Sparsification - a technique for speeding up dynamic graph algorithms. *33rd Symp. on Foundations of Computer Science*, pages 60–69, 1992.

[70] R. K. Ahuja T. L. Magnanti J. B. Orlin. *Network flows.* Prentice-Hall, 1993.

[71] C. Palmer and A. Kershenbaum. An approach to a problem in network design using genetic algorithms, 1995.

[72] Vern Paxson and Sally Floyd. Wide area traffic: the failure of Poisson modeling. *IEEE ACM Transactions on Networking*, 3(3):226–244, 1995.

[73] Judea Pearl. *Heuristics: intelligent search strategies for computer problem solving.* Addison-Wesley Longman Publishing Co., Inc., 1984.

[74] C. Perkins. Ad hoc on demand distance vector (aodv) routing. IETF, Internet Draft, draft-ietf-manet-aodv-00.txt, November 1997.

[75] C. E. Perkins and A. Myles. Mobile IP. *Proceedings of International Telecommunications Symposium*, pages 415–419, 1994.

[76] Charles E. Perkins. *Ad Hoc Networking.* Addison-Wesley, 2000.

[77] O.R. Oellermann L.W. Beineke R.E. Pippert. The average connectivity of a graph. *Discrete Mathematics*, Volume 252(1).

[78] F. P. Preparata and M. I. Shamos. *Computational Geometry An Introduction Springer-Verlag.* Springer-Verlag, 1985.

[79] R C Prim. Shortest connection networks and some generalisations. *Bell Systems Technical jrnl*, pages 1389–1410, 1957.

[80] Qinetiq. *Internal Qinetic classified communication 1.* N/A, 2000.

[81] Qinetiq. *Internal Qinetic classified communication 2.* N/A, 2000.

[82] Qinetiq. *Internal Qinetic classified communication 3.* N/A, 2003.

[83] Qinetiq. *Internal Qinetic classified communication 4.* N/A, 2003.

[84] Qinetiq. *Internal Qinetic classified communication 5.* N/A, 2003.

[85] Rajmohan Rajaraman. Topology control and routing in ad hoc networks: a survey. *ACM SIGACT News*, 33(2):60–73, 2002.

[86] Ram Ramanathan and Regina Hain. Topology control of multihop wireless networks using transmit power adjustment. In *INFOCOM (2)*, pages 404–413, 2000.

[87] R. S. Barr J. P. Kelly M.G.C. Resende and W. R. Stewart. Designing and reporting on computational experiments with heuristic methods. *Journal of Heuristics*, 1(1):9–32, 1995.

[88] J. D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithm. *In Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 93–100, 1984.

[89] WJ Cook WH Cunningham WR Rulleyblank A Schrijver. *Combinatorial Optimization (Wiley-Interscience Series in Discrete Mathematics).* Wiley Interscience, ISBN: 047155894X, 1998.

[90] Chien-Chung Shen, Chavalit Srisathapornphat, Rui Liu, Zhuochuan Huang, Chaiporn Jaikaeo, and Errol L. Lloyd. Cltc: A cluster-based topology control framework for ad hoc networks.

[91] W Simpson. The NP-completeness of edge-colouring. *SIAM J. Comput 10:718-720*, 1981.

[92] W Simpson. The point-to-point protocol. *Internet Request For Comments 1661*, 1988.

[93] M. I. Skolnik. *Introduction to Radar Systems.* McGraw Hill, 2001.

[94] N. Srinivas and Kalyanmoy Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.

[95] W. Richard Stevens. *The Protocols TCP/IP Illustrated Volume 1.* TCP/IP The Protocols Applications and Impementation. Addison-Wesley Pub Co, 1994.

[96] Andrew S. Tanenbaum. *Computers Networks.* Prentice Hall, 1996.

[97] A.V. Goldberg R.E. Tarjan. A new approach to the maximum-flow problem. *Journal of ACM (JACM)*, 35(4):921–940, 1988.

[98] R. E. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1:146(4):146–160, 1972.

[99] S.D. Milner J.E. Wieselthier R.R. Iyer K. Chandrashekar S. Thakkar and G.D. Nguyen. Scalability of dynamic wireless tactical networks. *MILCOM 2001*, Washington, DC.

[100] T.R. Jensen B. Toft. *Graph Coloring Problems.* Wiley Interscience Series in Discrete Mathematics and Optimization. John Wiley Sons Inc, 1995.

[101] Y. Tseng, Y. Chang, and B. Tzeng. Energy-efficient topology control for wireless ad hoc sensor networks, 2002.

[102] Jean Walrand and Kallol Bagchi. *Network Performance Modeling and Simulation.* Taylor and Francis, Inc, 1998.

[103] W. Wang, D. Tipper, B. Jaeger, and D. Medhi. Fault recovery routing in wide area packet networks. *Proceedings of 15th International Teletraffic Congress, Washington, DC, June.*, 1997.

[104] Roger Wattenhofer, Li Li, Paramvir Bahl, and Yi-Min Wang. Distributed topology control for wireless multihop ad-hoc networks. In *INFOCOM*, pages 1388–1397, 2001.

[105] H Whitney. Congruent graphs and the connectivity of graphs. *American Journal of Math*, 54(3):150–168, 1932.

[106] M. H. Wright. Optimization methods for base station placement in wireless applications. In *Proceedings of 1998 Vehicular Technology Conference*, volume 89, pages 11513–11517, 1998.

[107] J. Xu, S. Chiu, and F. Glover. A probabilistic tabu search for the telecommunications network design. Combinatorial Optimization: Theory and Practice 1 (1996), 69–94.

[108] H. Zimmerman. Osi reference model-the iso model of architecture for open systems interconnection. *IEEE Transactions on Communication*, 28(4):425–432, 1980.