Deterministic Ethernet in a Safety Critical Environment

By Yuen Kwan Mo

This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

School of Engineering

THE UNIVERSITY OF
WARWICK

September 2014

For the future of Zero Latency Networking

Table of Contents

List of Figures

Lists of Tables

List of Abbreviations

ACK – Acknowledgement.

AEEC – Airlines Electronic Engineering Committee

AFDX – Avionics Full-Duplex Switched Ethernet

AN – Acknowledgement number

ARINC – Aeronautical Radio, Incorporated

ARQ – Automatic Request

ARTT – Average Round-Trip Time

ASTERIX – All-purpose STructured Eurocontrol suRveillance Information eXchange.

ATC – Air Traffic Control

ATM – Synchronise Transfer Mode

BER – Bit Error Rate

CAN – Control Area Network

CART – Classification and Regression Tree

CDP – Cisco Discovery Protocol

CSMA/CD – Carrier Sense Multiplex Access with Collision Detection

DoS – Denial of Service

DSP – Digital Signal Processor

E1 – European Carrier

EUC – Equipment under Control

EIGRP – Enhanced Interior Gateway Routing Protocol

FEC – Forward Error Correction

FTP – File Transfer Protocol

GA – Genetic Algorithm

HDLC – High Data Link Control

ICMP – Internet Control Message Protocol

IEC – International Electrotechnical Commission

IGMP – Internet Group Management Protocol

IMA – Integrated Modular Architecture

IP – Internet Protocol

IPv4, IPv6 - Internet Protocol version 4, respectively 6

ISO – International System Organisation

LAN – Local Area Network

LOPA – Level of Protection Analysis

LRU – Line Replaceable Unit

NextGen – Next Generation Air Transportation System

NTO – Network Traffic Oscillator

OFDM – Orthogonal Frequency Division Multiplexed

OSI – Open System Interconnection

OSPF – Open Shortest Path First

PPoE – Point-to-point Ethernet

RIP – Routing Information Protocol

RTP – Real Time Protocol

SDH – Synchronous Digital Hierarchy

SESAR – Single European Sky ATM (Air Traffic Management) Research

SIL – System Integration Level

SN – Sequence Number

SNR – Signal to Noise Ratio

SONET – Synchronous Optical Networking

SRTT – Singularity Round-Trip Time

STM – Synchronous Transport Module

SVM – Support Vector Machine

SWIM – System Wide Information Management

TCP – Telecommunication Control Protocol

TCP/IP – Transmission Control Protocol/Internet Protocol

TDM – Time Division Multiplexing

TDMA – Time Division Multiple Access

TTE – Time Trigger Ethernet

UDP – User Datagram Protocol

VLAN – Virtual Local Area Network

VOIP – Voice over IP

ZOH – Zero Order Hold

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Dr Mark Leeson, for his support, guidance and encouragement throughout this research, and for allowing me to grow as a communication technology researcher. Your advice on research has been invaluable. I would also like to thank my second supervisor, Prof Roger Green for your brilliant comments and suggestions.

I would also like to thank Dr Christos Mias and Prof Declan Bates for serving as my panel committee members, and providing useful comments and suggestions throughout my project. Special thanks go to my mother and father for all the support they have given me throughout this project.

I wish to express my gratitude to the Engineering and Physical Sciences Research Council for providing me with the funding to complete this study, my industrial partner (FTI) who provided communication traffic data to support my research, and OPNET Technologies Inc for providing me with an academia license for their software used in the network simulation. I would also like to thank all of my colleagues in the Warwick Communication Systems Laboratory (ComSysLab), who provided emotional support during difficult times in my research. I would like to thank all of the staff in the University of Warwick who helped with the success of this research.

And finally, I would like to thank Dr. Daciana Iliescu and my good friend Michael Edmonds for assisting me to proof read and format this thesis.

Declaration

This thesis is submitted in partial fulfilment for the degree of Doctor of Philosophy under the regulations set out by the Graduate School at the University of Warwick. This thesis is solely composed of research completed by Yuen Kwan Mo, except where stated, under the joint supervision of Dr. Mark S. Leeson and Prof. Roger J. Green between 2010 and 2014. No part of this work has been previously submitted to any institution for admission to a higher degree.

Yuen Kwan Mo

September 2014

Thesis Abstract

This thesis explores the concept of creating safety critical networks with low congestion and latency (known as critical networking) for real time critical communication (safety critical environment). Critical networking refers to the dynamic management of all the application demands in a network within all available network bandwidth, in order to avoid congestion. Critical networking removes traffic congestion and delay to provide quicker response times.

A Deterministic Ethernet communication system in a Safety Critical environment addresses the disorderly Ethernet traffic condition inherent in all Ethernet networks. Safety Critical environment means both time critical (delay sensitive) and content critical (error free). Ethernet networks however do not operate in a deterministic fashion, giving rise to congestion. To discover the common traffic patterns that cause congestion a detailed analysis was carried out using neural network techniques. This analysis has investigated the issues associated with delay and congestion and identified their root cause, namely unknown transmission conditions. The congestion delay, and its removal, was explored in a simulated control environment in a small star network using the Air-field communication standard. A Deterministic Ethernet was created and implemented using a Network Traffic Oscillator (NTO). NTO uses Critical Networking principles to transform random burst application transmission impulses into deterministic sinusoid transmissions. It is proved that the NTO has the potential to remove congestion and minimise latency. Based on its potential, it is concluded that the proposed Deterministic Ethernet can be used to improve network security as well as control long haul communication.

Publication Associated with this Research Work

Book Chapter

Y. K. Mo, M. S. Leeson, R. J. Green, “Deterministic Ethernet Using a Network Traffic Oscillator”, in IAENG Transactions on Engineering Technologies, Springer, 2015,

Conference Papers

Y. K. Mo, M. S. Leeson, R. J. Green, “Effective Frame Switching using a Network Traffic Oscillator”, *in The International Conference of Information Security and Internet Engineering (ICISIE), IAENG, WCE*, London, U.K. July 2014

Y. K. Mo, M. S. Leeson, R. J. Green, “Network Traffic Oscillator for Safety Critical Networking”, *in 9th International Symposium on Communication Systems Networks and Digital Signal Processing , IEEE/IET*, CSNDSP, Manchester, U.K. July 2014

Y. K. Mo, M. S. Leeson, R. J. Green, “Intelligent optical Network Traffic monitor design”, *in 14th Anniversary International Conference on International Transparent Optical Networks, IEEE, ICTON* , Warwick, U.K. July 2012

Patent

University of Warwick, “Controlling Packet flow in a network” GB1411366.6, June 26, 2014

Chapter 1 – Introduction

As a Safety Critical System (SCS) is upscaled from a localised, self-contained application to multiple applications interconnected over a large area, the role of the communication network is crucial in maintaining the reliability, safety assurances and protection of the whole system. The subject of this thesis is the study of the network requirements for **real time communications** in a **safety critical environment** and the realisation of a Deterministic Ethernet which would fulfil those requirements.

On one hand, Safety Critical Systems (IEC 81508 [1]) are more relevant to the safety regulation of sensors, for example in chemical production and machine operations, where the malfunction in operation could lead to harmful events or even loss of life. The management of SCS refers to two processes, namely prevention and failsafe. An example of prevention is Safety Caution, an operation procedure to alert and evacuate the premises (such as alarms), and failsafe includes the Protection subsystem (damage control over the hazard) [2].

On the other hand, current real time protocols (RTP) for communication are designed by communication standards bodies for regulating Ethernet networks and do not specify safety critical features. In this sense, they are not exactly fit for this purpose, because the direction of Ethernet technology design has been focused on compatibility and flexibility issues, therefore it contains a fundamental weakness, namely lack of control over the traffic.

The work presented here aims to address this weakness and create a Deterministic Ethernet which supports real time, safety critical communications.

1.1 Research Motivation

This project involves a study carried out on a safety critical system for airfield communications. This type of system is managed by the Single European Sky Air Traffic Management Research (SESAR) [3]. SESAR and NextGen (Next Generation Air Transportation System) are standards committees for integrated air traffic design, management and maintenance in the Europe and the United States respectively [4-6]. Among many topics addressed by SESAR, two areas are relevant to the current project, namely Network Technologies and Communication Applications.

Network Technologies (network, data-link and physical layer of the OSI [7]) are the backbone of all inter-communication systems, both in connectivity and compatibility. Communication Applications include a large number of complex and diverse applications: Aeronautical Information Systems and Management, airfield operations, antennas and radars [8], Air Traffic Control and Mobile Control Towers [9], avionics [10], communications control, communication with voice [11], ground-to-ground air traffic data networks, flight data processing systems, integrated telephone systems and service, satellite navigation networks and surveillance systems.

Network Technologies and Communication Applications have conflicting operating requirements. From an application to network point of view, everything is optimised and operates within the computation algorithm design of the individual application, with no feedback to the network, whereas the network optimisation sometimes occurs in the detriment of the individual applications. In this case monitoring from the network perspective becomes difficult, with multiple applications arriving and dropping into the network.

The approach of this research is to rethink this conflicting approach and to provide a radical and optimal solution for both network and application. The first step of my research is to design a real time communication monitoring system to observe the congestion delay problem in a real time network. As SESAR plans to expand both air traffic and air traffic communication systems [3], monitoring would be a key solution for identifying implementation and safety issues in data communication areas. A monitored communication system leads to better risk assessment and control. This research will tackle the on-going challenge of monitoring such a system by managing data structure, mining and clustering, and ultimately creating traffic predictions about the safety critical communication.

Official telecommunication standards, such as the OSI model, give rise to a more independent and diversified approach when designing parts of a network [12]. Traditionally, hardware was directly connected to other hardware operating in a point to point network using Serial/Parallel links [13]. With the introduction of the OSI the networking task has been divided into seven mutually exclusive segments, called layers, which communicate through well defined interfaces. Each layer provides distinct services concerning either the physical medium technology (physical layer), physical hardware address (data-link layer), addressing (network layer), flow control (transport layer), peer-to-peer communication (session layer), security encryption (presentation layer) and finally applications. The OSI model is popular amongst application developers [7], network administrators and communication engineers. Each role focuses on only one aspect of this network system. In general, tasks are catalogued by areas which fit within those roles, but any tasks that fit more than one role could be pushed to any of these areas or are inadvertently overlooked.

In reality, a change in one layer has a knock-on effect over other layers, for example protocol design choices affect hardware technology choices, and sometimes create problems in other layers without having the means to correct them. For this reason, this research considered traffic optimisation from the separate perspectives of the network and the applications (i.e. within the respective OSI layers), but concluded that a cross-layer approach to optimisation of the entire network would give better results. However, a detailed investigation on an entire network has never been previously carried out due to the complexity of organising information in a network.

Traditional industrial networks are relatively small scale, often with only a few machines being networked together to perform a small task. In consequence, network errors are of relatively low importance compared to other problems in a factory (e.g. a machine breaks down) and also localised [14]. As a company expands to a global scale, connectivity inside local premise is not enough and networks are expanded to connect other premises in cities, countries and even across multiple continents. As the scale of operation becomes large, so does the scale of the previously small errors found on local premises.

This thesis tackles the challenge of mapping a complex network operation between applications, protocols and network hardware technologies, and provides a safety critical network transparency over its operations [15], as it is especially critical in the case of SESAR [16]. Critical Networking takes full advantage of a detailed investigation, from modelling high speed switching over communication technologies to simulating the effect of using such design.

In this study, the application is a time critical transmission from the radar to the control tower. The radar transmits over Ethernet protocol, Internet Protocol (IP), User Data Protocol (UDP) and All Purpose Structured Euro-control Surveillance Information

Exchange (ASTERIX Presentation Protocol) [17]. ASTERIX is the presentation protocol set by the Euro-control radar communication authority [11, 18-20]. The radar will be later integrated into SESAR plans for a System Wide Information Management (SWIM) network model. This study examines therefore time critical application transmissions over the SESAR Cloud.

The most important aspect in Safety Critical Communication is latency. This is the response time between a sender and a receiver, measured by the time (delay) for a transmission to gain a response. The scale of current communication networks has increased in complexity for managing this level of latency securely and, especially when networking becomes more automated, any small delay can grow into a large one due to network mismanagement and miscommunication in a safety critical network can be catastrophic.

Traditionally all automated networks were relatively small, and the communication equipment produced many unexpected errors from poor quality of service amongst service equipment. Extra packet diagnostic messages were required to be embedded in a packet to make each communication unit more transparent for recovery and diagnostic. The Ethernet protocol has been created to unify the basic overhead packet structure for all automated communication. The concept of Internet Framework was created to expand this structure from an end-to-end point perspective to support not only user messages and machine communication, but also messages between two pieces of networking equipment [21]. In order to achieve inter-operability and error control, communication instructions and information (overheads) are added to every automated unit of communication (packet) transmitted [14]. Unfortunately, these overheads reduce the accessible bandwidth for other safety critical applications in a network, leading to queued, blocked or dropped packets.

Modern communication networks offer routinely physical bandwidths in the Gbps and Tbps range [22], to satisfy the need for high speed data exchange and processing for the increasing number of data-intensive applications. At the same time, the Internet paradigm makes it the most widely accepted infrastructure in the world, making the Internet Protocol (IP) and IP-based transport protocols, such as Telecommunication Control Protocol (TCP) and User Datagram Protocol (UDP) hold sway in the transmission process [23]. IP-based protocols also support sockets, which have become the de facto interface standard [24]. It is thus not surprising that the incorporation of critical networking applications, such as airfield communications, into the IP family has been proposed in order to standardise equipment and interfacing. However, in their current incarnations, IP-based protocols present a barrier to maximum network throughput because of their substantial overheads [25].

Despite both long-standing (e.g. proxy caches [26]) and much more recent (e.g. optimising resource allocation [27]) network-based efforts to improve matters, there is still room for improvement. To date, major effort has been expended to reduce protocol overheads by the production of new high-speed user-level protocols [28], or by optimising IP-based protocols [29]. Although the first of these provides new light-weight transport protocols, their simplicity and new interfacing requirements mean that they cannot provide straightforward compatibility with IP networks. Moreover, despite the potential of the second approach, in this thesis it is proposed that efforts should return to the areas of traffic shaping and flow management that seem to have largely disappeared.

1.2 Thesis Contents

This work covers important aspects of Critical Networking. Following the description of the need for creation of a Deterministic Ethernet addressed previously in Chapter 1, the thesis includes in Chapter 2 a review of the available networking paradigms and technologies. A short description of the Ethernet, as the fundamental technology used throughout the thesis, and the relevant protocol descriptions as defined by the OSI are given. The chapter also includes textbook material on generic Safety Critical Systems and specific standards used in avionics to introduce the context of the safety critical airfield radar application for which the Deterministic Ethernet has been devised. General information extraction and specific network monitoring techniques are also revised.

In Chapter 3 a detailed analysis is carried out using neural network techniques to discover the common traffic patterns that cause congestions. The most important design issue has been identified as ‘removal of unknown transmission congestion delay’. The congestion delay is explored further in Chapter 4 in a control simulation environment in a small star network using the Air-field communication standard. A solution for the Critical Networking problem is proposed in the remaining chapters. A Deterministic Ethernet is created and implemented using a Network Traffic Oscillator (NTO) in Chapter 5.

The NTO is inspired by traffic shaping flow controllers for traffic regulation, which existed in Asynchronous Transfer Mode (ATM) [6] but were subsequently forgotten in the design of Ethernet. This critical networking flow controller no longer uses overhead to regulate traffic but rather a traffic shaper scheme with deterministic reserved bandwidth. Different services can sub-divide the bandwidth to optimise their

service rate for the user rather than by the approximation of services estimating the network resources and limitation for number of network devices [3] and overheads [1]. The critical network then assigns each application an arbitrary transmission frequency with allocated deterministic payload based on the type of protocol, its estimated arrival rate and its payload size. In short, this method allows higher connectivity by removing unnecessary overheads and ensures better arrival rate of information through network bandwidth re-allocation and optimization. It also promotes higher transparency, as each application can be cross-correlated not just by protocols headers, but also by its transmission frequency even across multiple technologies, standards and protocols.

Applications of the proposed Deterministic Ethernet are given in Chapter 6. Firstly, the advantages brought in the field of long haul communication are shown, followed by possible uses in network security. The thesis ends with conclusions and further thoughts about future developments of critical networks in Chapter 7.

Chapter 2 – Literature Review

This chapter gives the background for current network technologies and a review the ongoing design issues created by the over-compartmentalisation introduced by the OSI layer design within the Ethernet protocol. It describes the required avionics standards and the knock-on effect that the network design has on modern safety critical networks from factories to avionic communication systems. It also addresses information extraction and networking methodologies, which are used in the thesis to tackle these modern network problems.

2.1 History of Network Design

The history of communication networks sparked from the need to reduce the number of cables connected to every machine. This created the need for shaping traffic and payload. The two alternative perspectives for network management design are viewed as application or maintenance based [30], see Figure 2.1.

In an application based network each application controls and directs traffic to the destination [31], for example direct dialling in a telephony network. A maintenance based network creates a virtual circuit link; much like an operator based telephone system, a complete circuit is setup before connecting a user to a call [32]. This process however, only uses a small amount of the channel's capacity, whilst blocking off any other calls occupying the same channel. As digital data communication use channel multiplexing and the rate of data communication exchanged (bandwidth) far exceeds

the arrival rate of application data, application based networks are far more popular due to their high flexibility and availability to accommodate additional data from other protocols. This additional data, called overhead, includes sophisticated controls such as diagnostics, error correction, network organisation (networking layer) and recovery (transport layer) [33]. A Controller Area Network (CAN) [34] is a good example of a low data transfer rate service over a high bandwidth network, where information is generally exchanged in small units and there are many types of flags to signal the type of packet, its creation and its travelling time. The CAN service is still an application based protocol, which deviates from the popular IP and transport layer based network.

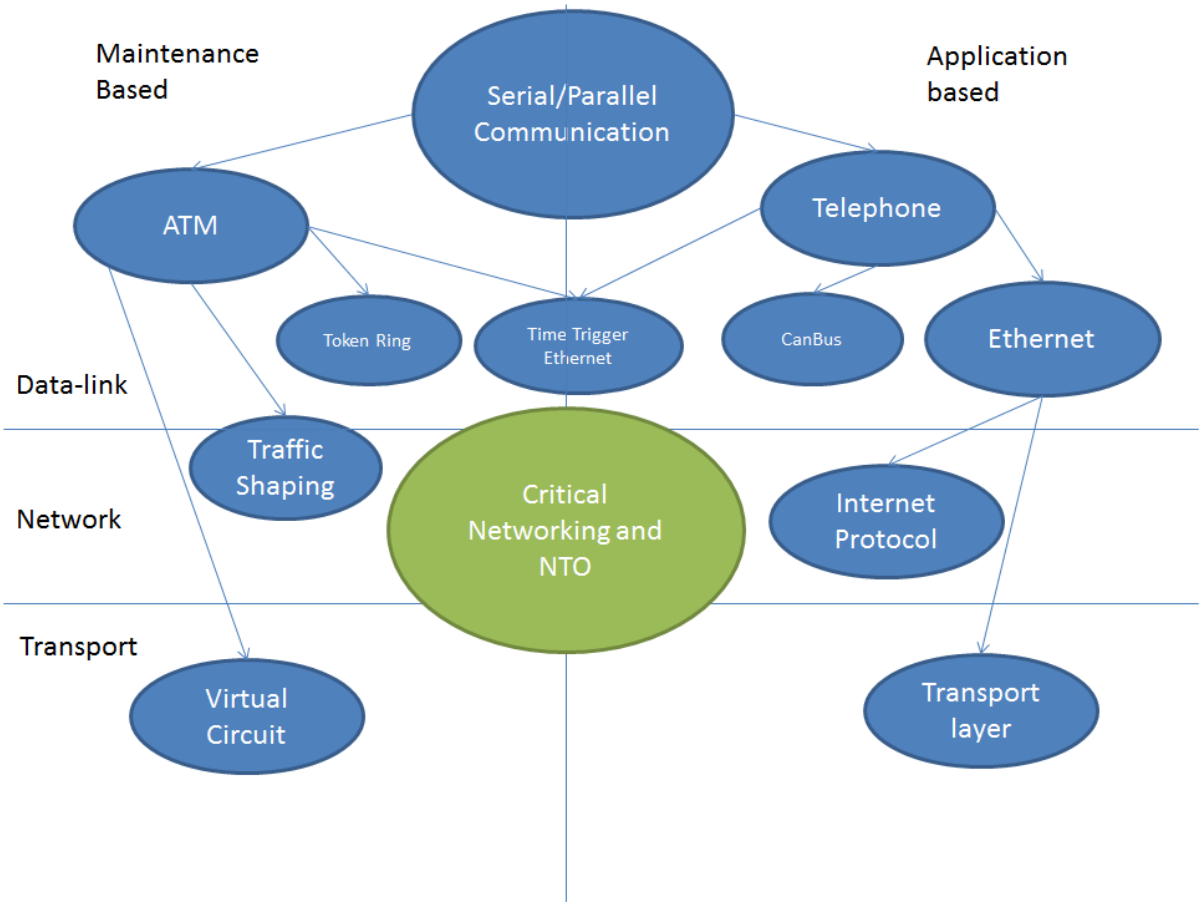


Figure 2.1 The re-direction of networking from maintenance to application based.

A network maintenance based communication uses Asynchronous Transfer Mode (ATM). Although it is inflexible for supporting multimedia services, its critical approach of controlling traffic had led to almost zero network traffic congestion [35].

There are many attempts to create a true hybrid system where the user service would have the flexibility of application based networking, while keeping the critical maintenance similar to ATM. Some examples are Time Trigger Ethernet (TTE) [36, 37], CanBus [38] and ATM style Token Ring [39]. The common element is the creation of micro traffic managing techniques embedded in the Data-link. These systems perform exceedingly well in networks with high available bandwidth, but lack the external control to handle traffic when the bandwidth is subject to availability. Two alternatives to traffic shaping exist [37], namely forcefully removing payload to handle priority payload in a maintenance based network, and introducing softer computation based routing algorithms [40] for an application based network. Finally, the main criticism of application based networks is that it doesn't necessarily guarantee the first payload transmission will reach the destination intact. An external transmission control, known as the transport layer, is designed to handle miss-transmission [33]. A maintenance based network, such as VLAN (Virtual Local Area Network), has virtual circuits [41] to guarantee every transmission at the cost of application inflexibility.

2.2 Protocols and the OSI

The OSI (Open Systems Interconnection) model is reference model for how applications can communicate over a network. It consists of seven layers which can be further subdivided into either Interoperability and Connectivity or End-to-End Connection and Chained Layer, shown in Figure 2.2.

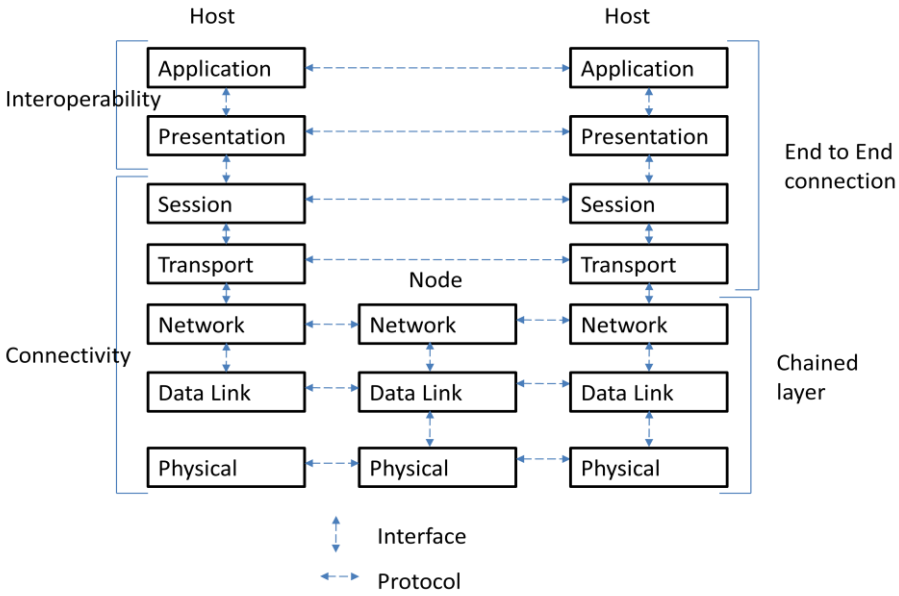


Figure 2.2 The 7 layers of OSI model [2]

The Interoperability group are the layers for the interpretation of the data into information by the computer while the Connectivity group is for the transfer of data without errors. An End-to-End connection is explained as one device connecting to another, while a Chained Layer concerns the transit of data in a network.

The seven layers of the OSI model are:

Physical – Physical medium of the network (radio, optical and cable)

Data link – Physical addressing of the network (MAC address)

Network – Logical addressing, automatic request, frame control

Transport – Flow control of the network (end-to-end connection, reliability)

Session – Security control of the network (inter-host communication)

Presentation – Translation between encryption to information (data representation)

Application – Service of the program in the network.

Traditionally, an application based network uses the division of labour in network design to divide the task, focusing on different aspects. A communications engineer designs the network from the physical layer using performance based objectives such as bandwidth and bit error rate (BER), a network engineer designs a resilient network based on application demands (recoverable connection, multicasting), and an application engineer designs a service that is compatible with the network. This perspective is useful for task division and network planning, but when the network becomes large and complex, true network management becomes much more difficult without undertaking all aspects of the layer division in communication and networking at the same time, rather than independently.

2.3 Ethernet

The most widespread application based network, to the point of being the de-facto standard, is Ethernet. In 1970, Ethernet was developed by Xerox for Local Area Network (LAN), designed using a copper coax cable medium with a bus network topology [45]; bus topology networks generally suffer from communication collisions, as there is no control over medium sharing. Ethernet is a physical and data-link layer technology using a competitive computation method as opposed to a network centric organisation [46]. The term “Ether” treats the large network as a cloud. By nature, Ethernet cannot maintain simultaneous two-way transmission and thus when two senders transmit concurrently, a collision occurs. One way to tackle this problem is by CSMA/CD (Carrier Sense Multiplex Access with Collision Detection). Each transmitting node listens for messages transmitted from other sources before transmitting; if both senders transmit at the same time, they both stop, and a random counter from each counts down until it is ready to transmit. When there are many senders (two or more) within the same medium and channel, the chances of a collision are higher. Technology such as an Ethernet switch can break down a large network buses into point-to-point link and time slots allocated for any incoming message using Time Division Multiple Access (TDMA). Modern Machine Ethernet LANs use star network topologies [47] and remove much of the collision problem, but the competitive computation style protocol design still exists today. The common physical medium now uses four sets of twisted copper wire pairs and each node connects to the star network topology. The central network is replaced by a hub, switch or router; these devices change the behaviour of network traffic. A hub is a central port device comprised of

many repeaters, each of these connects two portions of smaller LANs by forwarding all data received by the repeater (broadcast). A switch comprises of multiple bridges, and each of these connects one LAN to the other intended LAN using a physical MAC address. Classic Ethernet modes set by the IEEE 802 standard [48] include; 10Base5[39], 10Base2[49], 10BaseT[48], 10BaseF[50] and 100BaseT [51, 52].

2.3.1 Ethernet Protocol

The physical layer and a data-link layer of the Ethernet protocol are referred to as an Ethernet frame. This has a large overhead, where information and network instructions are embedded within each packet transmitted. A packet transmission requires no additional knowledge about the network topologies or the underlying physical medium technology to transmit in this system. Each piece of network equipment adds its relevant overhead depending on the network layer on which the equipment operates [42]. Ethernet is highly compatible with the explosive expansion of information services within modern communication networks and there are many protocols that can utilise this level of technology by piggybacking. TCP/IP is a protocol that uses only four layers of the OSI model and it is popular for commercial networks. Some of the OSI layers are concatenated as shown in Figure 2.3.

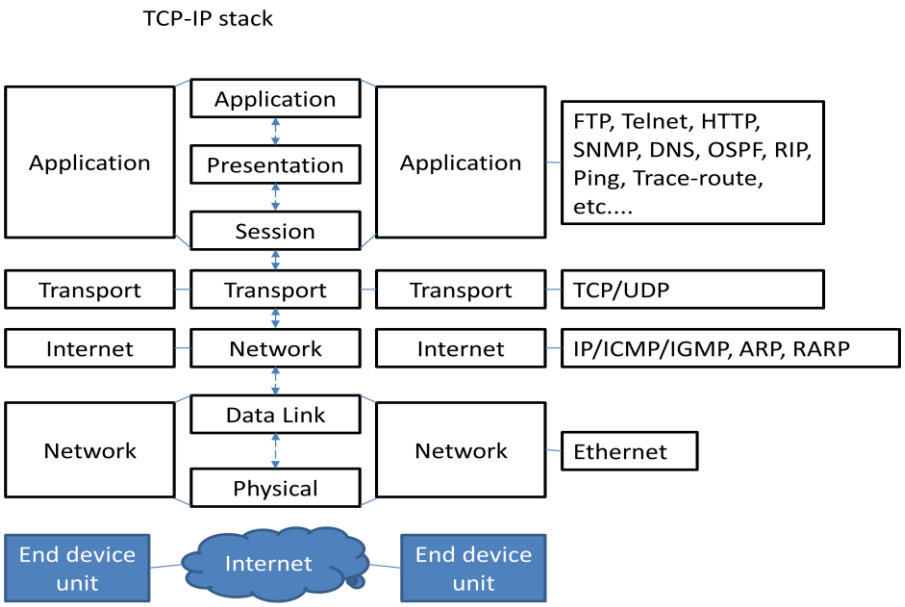


Figure 2.3 TCP/IP stack created in the OSI model with the supported protocols [2]

However, the disadvantage is the lack of conformity within packet transmissions for effective networking. Even when there are many standards and restrictions for packet size, length, type of overhead, there is no prior knowledge for the network to determine the actual payload size per packet. This unknown quantity that is payload size per packet has led to many networking problems. Different application services have increased the diversity of payload size per packet in a network, which increases the complexity for any effective networking scheme. In addition, each application has a different time interval between packet transmissions and different expected packet times of arrival. Subsequently, many packets are lost due to transmission timeout (the packet response time agreed between the transmitter and receiver has expired), and also due to poor routing decisions which led to more packets being dropped (packets blocked from switch and router due to buffer overflow). Each packet transmitted now has a recovery process (retransmission), and a transmission packet time control regulation system (the packet is deleted when it has exceeded its time-limit or hop

counter). These extensive levels of network packet control only reduce the problem, but do not prevent its occurrence. These networking problems are created from poor network management (queuing and serving), leading to more packet congestion in a network. The two types of transport layer flow management schemes inspired by a network transport iteration windowing process can be defined as connection or connectionless transport protocols [43] and it is the last line of defence for recovering missing packets. The two distinct characteristics are that the former uses feedback (acknowledgement between transmitter and receiver) to address congestion packet issues and the latter has no congestion management [44]. A connection oriented transport protocol is only suitable when there is feedback from the receiver to alter the level of transmission from the transmitter to suit the network capacity. A busy network cannot guarantee this level of service, creating vulnerability in the network design. Therefore all packets transmitted are always delivering the best effort service (the first packet transmission is not assured), despite these high levels of protocol control. Time-critical Ethernet packets have increased the complexity of networking by reserving time slots for a selection of applications, even when some of the application packets may arrive late. Priority Ethernet overhead [37] has been invented for queue jumping to reduce a selective packet delay, but still suffers from packet congestion where there are many priority packets stuck in a queue. These many levels of networking perspective have increased the level of complexity in network management and switching especially when the network is busy. In summary, a network is difficult to manage when there is a random traffic load in the system.

2.4 Real Time and Safety Critical Industrial Ethernet Systems

This section dives into the current protocol design of a real time critical network in a critical infrastructure. This critical infrastructure contains time-critical payloads (delay intolerant and error sensitive) which can cause issues, especially when this information is transmitted over mixed physical technologies in a commercial network.

In the 1980s, real time critical communication technology dictated application transmission patterns, and communication bandwidths offered very little degree of flexibility for network packets in terms of arrival times. In 1999, the Fieldbus international standard IEC 61158 had approved eight sub-classes of field-bus [53, 54]. These subclasses are:

- Foundation Fieldbus H1
- ControlNet
- PROFIBUS
- P-Net
- FOUNDATION fieldbus HSE (High Speed Ethernet)
- SwiftNet (withdrawn)
- WorldFIP
- Interbus

As inter-compatibility issues became dominant, these technologies become obsolete for a large network. Layer division [7] expands the freedom of movements for packet based traffic. Applications are no longer being restricted by network design constraints. By adopting the OSI layer model and dividing the transmission into seven segments, the structure is simplified.

In 1980 there were a variety of Fieldbus designs for safety critical Ethernets, to

be used in power plants and automated processor plants (Figure 2.4). Fieldbus technology divided automated plants into five levels; Factory level with computer aided design and manufacture, Planning level with planning computer, System Cell level with cell computer, Control Process level with closed loop controllers, and Sensor Actuator level with sensors and motors [55].

The common three sub-classes of Fieldbus are Profibus (XML tag base[56]), ControlNet and DeviceNet. However, as they do not interoperate amongst themselves, a gateway was designed to tackle the frame structure of each protocol [57]. The latest Fieldbus, called Modbus, has a serial bus like structure [58-61] and the portability is peer-to-peer oriented (transport data from point-to-point using TCP [62-65]). As Modbus operates in serial transmission, ZigBee technology is used to convert from serial Modbus to wireless [66]. As nodes and hosts increase, conversion from Modbus to Controller Area Network (CAN) is suggested [67]. Fieldbus Foundation High Speed Ethernet (FF-HSE) became the prime protocol for development and this Fieldbus was adapted to an open low cost time critical wireless Fieldbus architecture [68] . As the demand for the number of hosts increased in applications from automated plants to electronic car control systems using CANs, network IP addresses [46] were improved from 32 bit (IP version 4) to 128 bit (IP version 6). As the address data sizes increased, the frame designs originally used for DeviceNet and ControlNet became unsuitable and were replaced by Ethernet frames. The different status flags that originated from previous versions of DeviceNet and ControlNet, have been removed. Hosts and nodes of the network are given different IPv6 addresses.

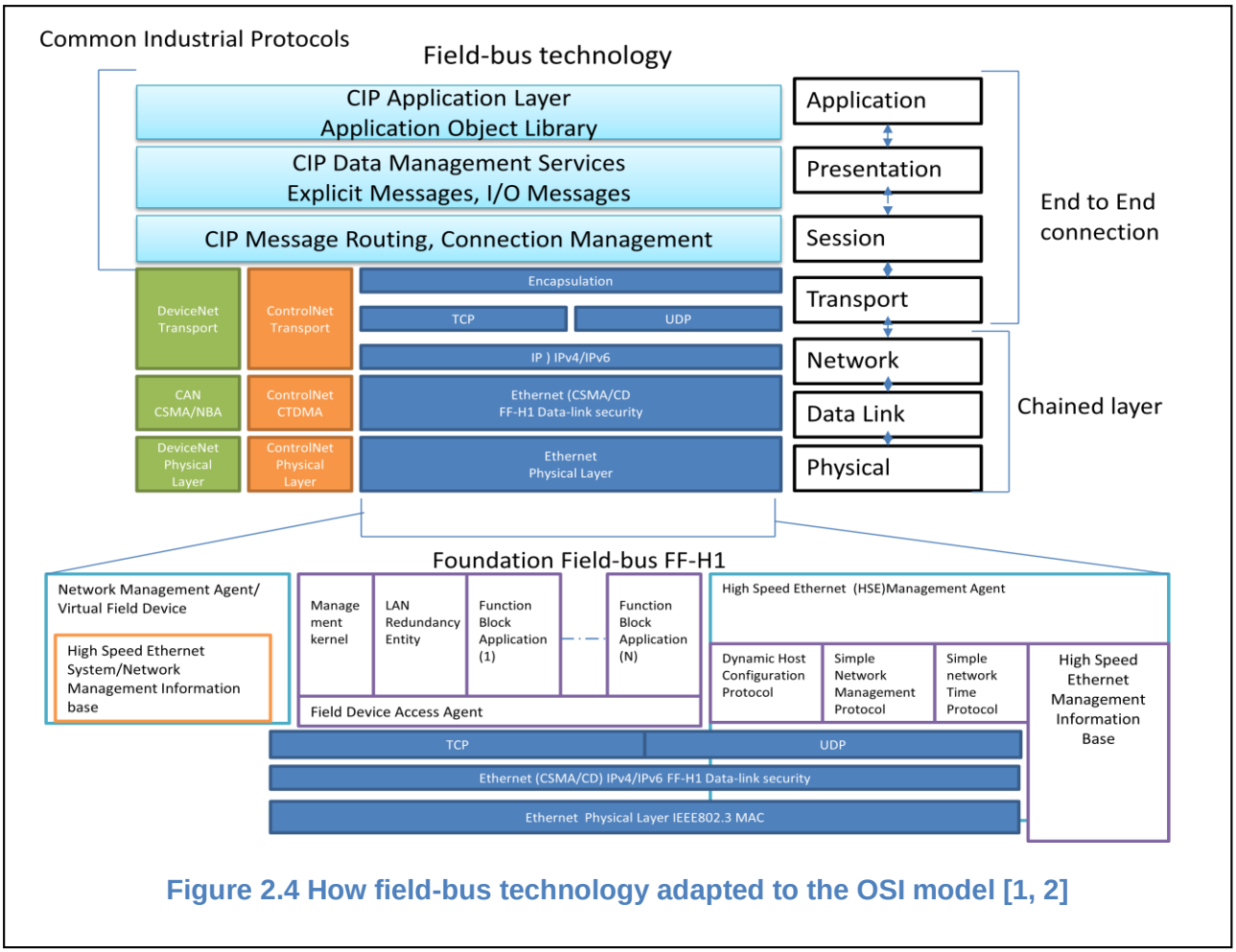


Figure 2.4 How field-bus technology adapted to the OSI model [1, 2]

A host group with a selected priority IPv6 address will have higher priority over transmission [53]. Foundation Fieldbus uses a protocol stack architecture similar to that of TCP/IP (Transmission Control Protocol/Internet Protocol) [14] to extend interoperability between current network equipment and older, serial-transmission based equipment in automated plants [14, 55, 69] . Rather than using Internet Control Message Protocol (ICMP), a Management Information Base was used to keep track of all the nodes and hosts in the network [14]. In Figure 2.2 the three stack technologies are compared with the OSI model and the top three layers have been divided according to the common industrial protocols. Hosts such as sensors transmit using User Datagram Protocol (UDP) whereas control modules employ TCP [53]. As Fieldbuses are generally used in small local area networks (LANs) where the

transmission has a determined status report (round-robin cycle [70]) no analysis, apart from a Fieldbus plant simulator[71], has been performed regarding its traffic control. In Avionics, some of the foundation Fieldbus technology was adapted as will now be discussed.

2.5 Ethernet Systems in Avionics

The Airlines Electronic Engineering Committee (AEEC) and the Aeronautical Radio Incorporated (ARINC) collaborate as the providers of telecommunications for avionics. ARINC's system is based on the Digital Federated Architecture, designed to communicate information to the Line Replaceable Unit (LRU AKA Black Box). The Federated Architecture contains flight management, communication management and analogue signal consolidation and conversion to digital data. Planes such as the Boeing 767, Boeing 757 and Airbus A320 were the first commercial aircraft to use a Digital Federated Architecture to extend the flight by wire ability. The ARINC 429 standard defines one hundred unique labels in a 32 bit data word (Overheads). Boeing 777 uses the Federated Architecture with Integrated Modular Architecture (IMA). This includes the Airplane Information Management System, Flight Management, Communication Management and Aircraft Condition Monitoring, with the LRUs being independent from the IMA in the previous design [72]. Avionics Full-Duplex Switched Ethernet (AFDX) comprises five OSI layers; Physical (ARINIC 664 pt 2), Data link (MAC Virtual Link), Network (IP ARINC 664 pt 3), Transport (UDP, TCP) and Applications (Secure, Avionics and Maintenance Application) [73, 74].

In contrast, the ARINC Spec 664.7 defines that the AFDX uses the Physical layer, with a switched Ethernet Full-Duplex system [75], removing the need for Carrier

Sense Multiplex Access with Collision Detection (CSMA/CD). The system is made deterministic using virtual links, redundancy bandwidth allocation and a priority system [75], as long as the physical layers are high performance (low bit error rate with error correction) and network traffic is under control. As the network size increases, traffic behaviour (multi-cast, uni-cast and payloads) changes dynamically and any problems in the network could cause serious aircraft instrument failure as all system are linked in the same network. Some work reported in the literature analyses network traffic using network trajectory measurement from host casting messages [72, 75] and generate these measurement into probabilistic model about the network operation parameters [76], such as the End-to-End delays inspired by network calculus [77, 78] and the latency from bandwidth allocation gaps. Other authors use simulation tools to model and measure the performance (delay, jitter) of an AFDX network [79-82]. Frame Management [83] and regulators such as a frame buffer [10, 73] have been proposed to stabilize burst traffic characteristic in AFDX. Increasing network size naturally increases the demands of traffic analysis and control, but there is little work in the re-designing of network architecture that prevents traffic overload. A deterministic avionic Ethernet system has two specifications for its connected devices, a protocol implementation conformance statement and a service implementation conformance statement. One provides a listing of all the protocols it can support while the other provides a list of services for increasing the inter-operability between suppliers. In the ARINC 429 standard, virtual links were used for direct unidirectional AFDX LAN communication. A predetermined bandwidth allocation gap was used to devise the minimum and maximum time between frames [74]. A predetermined maximum latency between transmitter and receiver is used to control the flow though the virtual links, thus increasing the bandwidth usage and shaping the traffic. Delivery messages are

additionally monitored and maintained using transmit and receive policing [80]. Redundancies are used in the physical layer, with full cyclic redundancy checking at the destination, switches and parallel connections. If a connection fails in either hardware, protocol or application, this system will accept a second copy of the signal. This level of management is handled in the application layer rather than in the network or the transport layer. As the avionic industry moves toward commercial off the shelf products there is an increasing demand for both effective network architectures [74] and network analysis tools [84, 85].

2.6 Avionic Communication Standards Authorities

Radar technology contributes substantially to the level of traffic in an avionic application. A standards committee called Euro-control provides a specification framework for radar [86]. The air traffic management research programme redefines the operation and management of air traffic into six categories [3]:

- Space-based navigation and integrated surveillance
- Digital communications
- Layered adaptive security
- Weather integrated into decision making
- Advanced automation of air traffic management
- Net-centric information access for operation

The general concept behind the Boeing air traffic management operation comprises five stages: Airspace management (which has a time scale of several years via flight planning), Flight operations management (up to one year), Flow management (up to

a day), traffic management (up to an hour) and separation management (up to 20 minutes) [3]. SESAR will also combine the airspace of different European countries into a single airspace, with a single regulator controlling airspeeds. Euro-control conducted simulations of different airspeeds [87] and also studies involving ground and aerospace communication manufacturers, aircraft manufacturers, airborne equipment manufacturers and air service navigation equipment providers. To support the growth in air traffic, the 4D (three dimension axis with aircraft speed) Trajectory Data Link Service provides a data communication service for Air Traffic clearances, moving from radar to air-to-ground communication systems [16, 88].

The technology for air-to-air communication [89] and air-to-ground communication will also be a combined telecommunication network using Voice over IP (VOIP). VOIP is currently supported by air service navigation providers in France and Germany for communication between aircraft and control towers in the airport [11]. As the level of information increases from the air-to-ground and air-to-air transmissions, there is an increased need for network capacity as well as demand for availability of the radio site [20]. The air-to-ground communication uses L-band Digital Aeronautical Communication System type 1 (L-band 960-1164 MHz) for surveillance capability [5] and physical layer performance [4]. This is a technology that combines Broadband Aeronautical Multi-Carrier Communication, and an Orthogonal Frequency Division Multiplexed (OFDM) radio based Technology (WiMAX) [4]. Real time Protocol (RTP) provides the standard for the Transport layer in the avionic industry. RTP 3550 is a protocol for End-to-End real time transmission [11] which was designed by the Internet Engineering Task Force in 1996 and then redefined in 2003. A new revised protocol, TCP/IP, may be employed to tackle the inter-operation between commercial

off the shelf products and other aeronautical communication networks in both NextGen and SESAR standards [44], but still holding onto the seven OSI layer architecture.

From 2011, as air traffic increases, SESAR also proposed to reduce the separation of airspeed between aircrafts [90]. A time-based spacing scenario depends upon the technology of the radar (scanning ratio) and factors such as: arrival separation (approach and diagonal radar minima) [91], aerodrome separation (departing, landing and runway) and departure separation (wake turbulence minima). Other authorities base the standard for aircraft separation on the weight of the aircraft [90]. Systems such as Enhanced Traffic Load Monitoring are also proposed by SESAR; this is an air traffic monitoring and management system which is based on workload and the capacity of the airspace and airport [9]. This increases the performance of a new type of radar for achieving the requirements. Additional work has been assigned to simulate the effects of Traffic Load Monitoring for air traffic control resources [9].

A merger of Air Traffic Management Systems from NextGen (US) with SESAR (Europe) into a service oriented SWIM architecture supported by Innovative Technology [18, 92] is planned. As SESAR unified the standard for Europe air traffic management, safety was one of the areas to tackle. The air traffic management infrastructure for safety [93] is specified as: Severity Classification Scheme, Assurance Level Allocation and Maintenance Intervention Assessment [94] . SESAR uses a barrier model for determining the safety service level of their system. As the aviation networks grow (addressing from IPv4 to IPv6) [95], security issues such as surveillance strategy over the usage of the network were also considered in the NextGen, SESAR and EUROCONTROL standards [8, 96]. The barrier protection is similar to the Layer of Protection Analysis (LOPA) in safety engineering [97]. The main three tier barrier is comprised of [93, 98]:

- Strategic Conflict Management
    - Airspace Design
    - Demand and Capacity Balancing
    - Trajectory De-confliction[1]
- Separation Provision
    - Coordination
    - Pilot Tactical De-confliction
    - Air Traffic Control Tactical De-confliction
- Collision Avoidance
    - Air Traffic Control Recovery
    - Pilot Recovery and Providence

2.7 Safety Critical Research

When a safety critical manufacturing machine requires connection to another safety critical machine for coordinated production, this link is considered a safety critical communication connection as the break-down in communication of one machine could lead to hazards and potential harm to human operators. A regulator body known as Safety Integrated Levels (SIL) has designed a framework for health and safety analysis [99]. Safety Critical Communication in SIL are considered at the service level, which is the probability of communication hardware break-down assessments, but not the assessment of network topology, technology and protocols affecting communication break-down. SIL is a standardisation for assessing a system's safety

[1] De-confliction, a method of avoiding mutual interference

levels. The relevant standards committee has created a generic framework (SIL determination) whereby all real time applications can be assessed; Diagnostics, Diversity, Specification, Design Process and Methods, Integration and Installation, Validation and Testing, Human Factors, Operation, Maintenance, Redundancy, and Reliability [100]. SIL contains the safety analysis of functionality and integrity aspects. The safety function is a hazard detection while safety integrity is hazard prevention, both are related to risk assessment [101]. The concept of risk in SIL is driven by the likelihood of the consequence and the severity of the consequence. SIL has created a standard for measuring a risk by the likelihood of consequence to model the system safety function [97]. Electrical/Electronic and Programmable Electronics are added as an additional protection layer to the safety system (LOPA). These electronics have a likelihood of breakdown less than the Equipment Under Control (EUC hazard analysis), thus reducing the overall likelihood of the consequence either by frequency of breakdown (EUC in high volume production) or probability of breakdown (EUC low volume production). Transparency of information in a safety critical system is a key issue addressed in SIL and this supports the development of a network data-link analytical tool that monitors the network integrity and performance of a safety critical telecommunication system.

Safety critical analysis is part of reliability engineering and involves analytical studies of a system breakdown with different modes [1]. These are fault tree analysis modes (the study of how one broken part impacts another), probabilistic models for each breakdown mode (calculating the likelihood of a single module breakdown in a multiple dependent system) and the mean time between breakdown and fault analysis mode (calculating the estimated time between each breakdown and the time taken to repair). If Ethernet was treated with this rigorous approach of assurance (using

payload collision and congestion payload as faults), it would be found to perform poorly in a busy network.

2.8 Safety Critical Ethernet Network

The simplest safety critical communication network connecting two machines together is a dedicated physical link. Assuming the link has high signal to noise ratio (SNR) and each operation is well below the link maximum data input capacity, the probability of breakdown between these two devices is in the electronic components inside the transmitter and receiver. This design is common, often found in safety critical application such as production plants and airfield communications around the 1980s, and is known as serial/parallel peer-to-peer communications [63].

Ethernet Networks [12, 102] have been introduced by Euro-Control for airport communication, and they are replacing these traditional peer-to-peer serial networks [46, 62], these systems are slowly making the crossover to half-duplex packet based systems [38]. A packet communication system, especially in a network, has high variation in transmission, and is undistinguishable in transmission characteristic (patterns). This thesis will investigate a method for monitoring packet based transmissions especially in network congestion, so that traffic and congestion patterns will be identified and the network model will be known, creating a more reliable and higher performance network than previously.

Although Ethernet is labelled only for physical and data-link layer technologies in the OSI model [7], SIL has been assessing the break-down level of buffer, bus or switch equipment instead of the network design. A Network layer protocol (IP) contains

additional control information for transmission in a network. This datagram layer packet is nested inside an Ethernet frame and often forgotten during the SIL assessment. The IP layer adds information about the type of network, the network construction (sub-network and their unique IP address), and the instruction for higher level equipment in a network such as a router, a bridge or a gateway, but SIL only measures the safety level of the individual equipment rather than the design framework [42]. The Quality of Service is maintained by the next layer of the OSI model. TCP and UDP were used in a safety critical network but these network designs were never questioned on the suitability of their configuration, only the generic break-down frequency was recorded [44].

In particular SESAR has devised a standard for the format of the Presentation layer for air traffic control communication equipment, named ASTERIX which includes surveillance data and binary messaging. This format is used as the Presentation layer for this research. ASTERIX has also proposed the use of UDP for real time critical application such as radar [17].

2.9 Information Extraction Techniques

Network calculus [42] is a modelling technique that simulate traffic behaviour of a peer-to-peer network. This model is an accurate model that describes the fundamental causality relationship between input and output of each individual OSI layer, through a greedy shaper and a convolution of components [103]. However, underpinning the fundamental flaws of modern safety critical networks requires the causality relationship of all layers across all traffic, which is difficult to model and understand by Network

calculus alone. Intelligent information mining, clustering and extraction were only used in relatively large databases related to modern networks [104-107], but results were inconclusive and localised to small network configuration remedies that temporary fixes the problem.

Currently, the literature in data communication traffic analyses only samples static packets of traffic history and represent the information with statistical analysis such as the Poisson arrival rate process. Poisson arrival process is used to determine the independent arrival rate of a packet buffer; in effect that would also determine the capacity of a network to reach a certain level of service [108]. Other studies in this field use statistics to determine stochastic (Discrete view) or continuous model analysis. The former study gives a static representation of a dataset in a statistical analysis; an example technique is called a self-organising data clustering which also uses a stochastic sample of a database for pattern recognition. Stochastic view models are based on these techniques to generate information [109, 110]. Monte Carlo simulation is one of the techniques that inserts a set of random inputs and summarises a set of outputs based on the recorded random inputs without the structure or framework of a system [111], which is an example of a continuous sampling models for a cause and effect relationship [112] to determine the condition of a network and forecast to predict outcomes, however the cause and effect model is often used in small networks [113] with low level of permutation results. As output datasets increase, data mining and clustering techniques become more relevant; cross discipline studies identified as top ten C4.5-5, k-Means [114, 115], SVM (support vector machine), Apriori [116], Expectation-Maximization, PageRank, AdaBoost, k-Nearest Neighbours, Naive Bayes and CART [117]. These techniques are used for data mining and clustering, which creates a framework, cataloguing datasets in groups by its patterns such as statistical

distribution and progression over a large complex dataset. SVM were chosen in this research because of support vectors identified in the dataset are scalable and mapped across many OSI layer traffic, which are imperative in unlocking the secret of network mismanagement. SVM uses a small set of the data (training database) to identify patterns in datasets. It is possible to use these techniques incorrectly leading to a miss-representation of the datasets, such as applying discrete statistical analysis when the dataset should be represented with a continuous statistical distribution[118]. As the database increases in size, a technique called Deterministic Annealing was chosen because it requires less training data than Neural Network Forecasting [119]. When a database contains multi-objective groups of information, Pareto-based Genetic Algorithms are used for modelling and grouping a multiple compromise solution [120], however network data do not require this level of complexity due to each traffic packet has been tag by their respective OSI layer.

2.10 Conclusion

A review of the literature indicates that Safety Critical Network traffic problems are poorly addressed and investigated in current practice, even when the traffic data itself are clearly labelled and separated by OSI layers. The intricacies of safety and industrial authority only scratch the surface as far as designing highly-flexible and supportive network for its certified applications is concerned. Current Ethernet design is detrimental to the SESAR plan of expanding all Safety Critical application under this architecture. This research determined valid methods and tools for analytical network condition monitoring, and overarching general traffic managing methods in real time

inspired by the analytical tool, will provide transparency for monitoring and controlling traffic in a safety critical application.

Chapter 3 – Network Effect

To support the main objective of this research to create a deterministic Ethernet supporting the SESAR-SWIM project (Connecting multiple airport communication systems together for time critical radar information), a network is created and monitored in a simulation environment. The environment reproduces a scenario for Great Britain, where there is an offshore wind farm in the North Sea to the North East of Scotland. The spinning turbine blades cause interference to the radar signal from Glasgow airfield, making detecting air targets (airliner) less accurate. Radar from Coventry is used to track air targets in the North and this information is sent back to the Glasgow airfield to help alleviate the problem.

This simulation uses a radar application which sends airfield targets periodically to the control tower, similar to that from the radar in Glasgow under the ASTERIX format. It also uses standard serial data gathered from the uncompressed data transmission and converts it to Ethernet format. This yields a standard periodic payload of 1300-1500 bytes. The compressed data transmission uses statistical compression by collecting three serial frames and multiplexes it to an Ethernet frame before transmitting. The routing uses point-to-point unicast algorithms, and is managed by TCP/IP. This simulation challenges the idea of using conventional Ethernet systems as proposed by SESAR, using instead packet multiplexing and packet compression (NTO) methods invented in this research. The concept of IP packet compression (conventional packet multiplexing) is found in [121]. This study takes a new perspective of network responses to packet/payload compression in a safety critical network. The general layout of the application is shown in Figure 3.1.

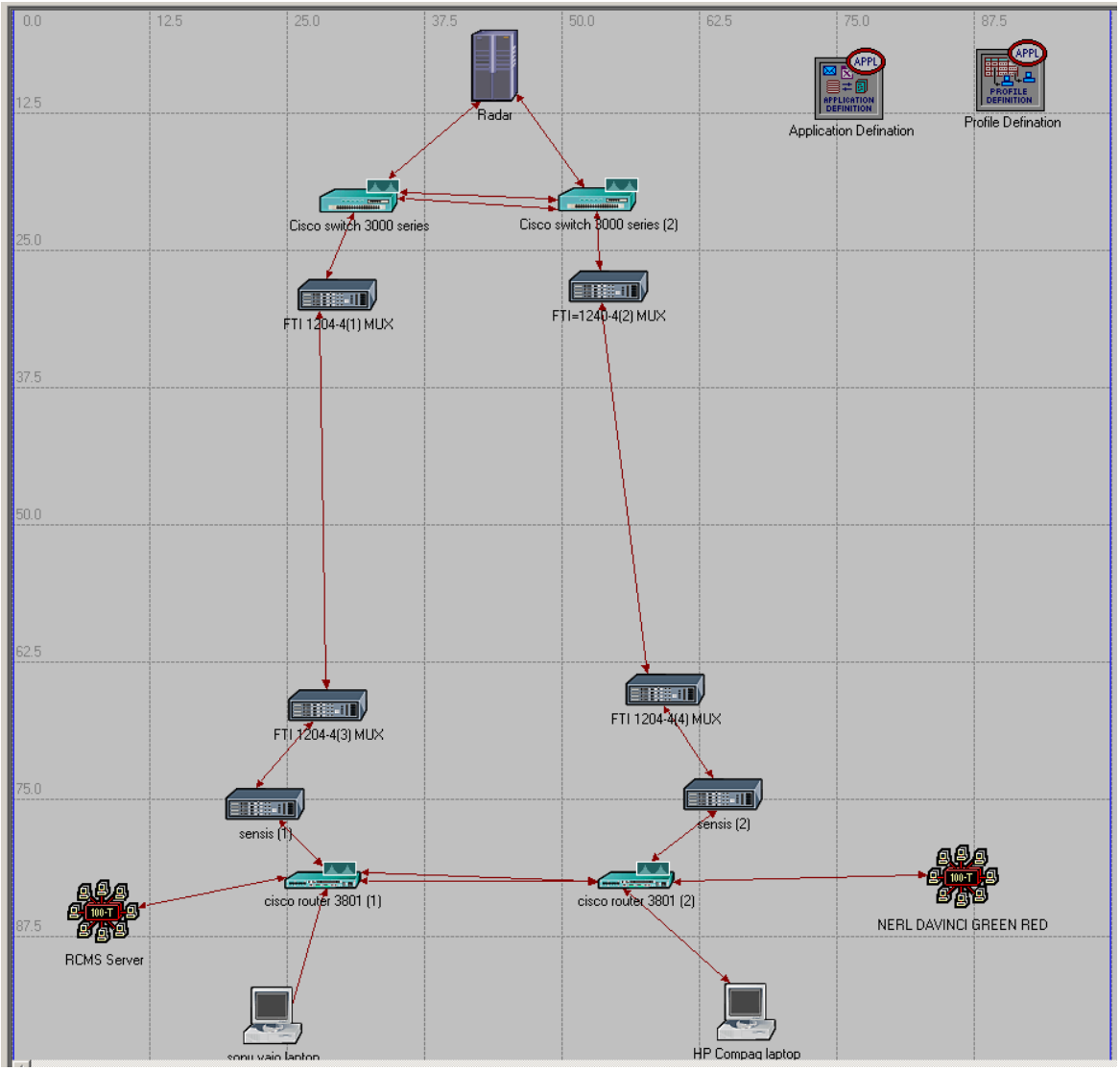


Figure 3.1 The Safety Critical Network Traffic Record setup and the network topology of Glasgow Airport

The on-going challenge comes from the fact that a telecommunication network is dynamically changing and constantly growing. Network calculus provides good models for describing parts of a network [42, 122] but has little to offer when it comes to increasing transmission efficiency, network switching, routing and management. Official communication and safety standards provide a good operating guideline of best practice for operators, but offer very little freedom for addressing network issues when they occur; the standard simply brushes past the effect of temporary incoming

network traffic congestion affecting the behaviour of a network over time [49]. This thesis' methodology ultimately tackles the complexity of creating a unified critical networking theory for targeting real time traffic models of an application based network. This network model will promote better practices in network operations and better insight to temporary network congestion behaviours.

3.1 Propagation, Congestion, Buffering and Retransmission Delay

The delay factor is the most important factor in safety critical networking, most information lost is not from a physical link high bit error rate or data corruption inside a buffer but by packets that exceed the travelling time, response time or from buffer overflow caused by excessive traffic. There are four aspects of delay that occur regularly inside a network:

- Propagation delay (also known as physical medium delay): this parameter is a constant and only scales with route distance; it represents the minimum traffic delay in the network and has a fixed value, unless there is a shorter route for traffic to be transmitted.

$$delay_P = \frac{distance}{propagation}$$

Equation 3-1

- Congestion delay: relative to number of extra packets in the system; scales with the service rate.

$$delay_C = \frac{extra\ load}{service\ rate}$$

Equation 3-2

- Buffering delay: caused by the limitation of bandwidth capacity on delivering the payload.

$$delay_B = \frac{payload}{bandwidth}$$

Equation 3-3

- Responsive delay: caused by the (transport layer) retransmission requirement of any missing data.

$$delay_R = nT_{retry\ time\ interval} \geq delay_p + delay_C + delay_B$$

Equation 3-4

As an example, when two nodes A and B (2 km apart) are connected to a copper medium bus network with a 10 Mbps link, the signal propagation in the medium is approximately $2x10^8$ m/s [123]. The trip time is 10 µs so the round trip time is 20 µs (Equation 3-1). The worst case scenario is when node A transmits first and node B transmits within the trip time interval; there is a window of approximately 200 data bits of collision opportunity (corrupting the smallest Ethernet packet of 512 bits) without any direct network control. The same trip time still applies as propagation delay regardless of bandwidth. The higher the bandwidth, the higher collision rate for the same packet size, therefore the minimum packet size is setup as 512 bits by the Ethernet standard [48]. Taking the number of bits and the propagation speed, a 10 Mbit/s transmission system will thus reach only approximately 2km using this default packet size. A 100 Mbit/s system would reach 200 m and a 1 Gbit/s system only 20 m. Increasing the default frame length by adding overhead padding allows a node to transmit greater distances and avoid high collision rates. For instance, using the same example above, an Ethernet frame of 4096 bits, can reach 1024 km whilst maintaining

one collision frame rate. Effective Ethernet frame transmission comparing padding to actual payload is low using this padding method. Frame bursting [124] is a concept that joins multiple frames in an end-to-end fashion before releasing them to the medium, extending the Ethernet frame size without the use of overhead. This subsequently increases the overall data throughput efficiency to 12% (dividing the 64 bit preamble with 96 bit of inter-frame gap), however this process is carried out by a statistical packet arrival rate in the Data-Link layer only, and could potentially cause other network delays in the system due to excessive queuing (Equation 3-2) and traffic buffering (Equation 3-3), hindering the packet transmission response time between transmission and receiving, and potential loss of transmission thus requiring retransmission (Equation 3-4). Dedicated links for transmitting and receiving (uplink/downlink) were used in the past to prevent these issues [12], but current SWIM architecture does not allow for this. Ethernet can be further extended to other technologies such as SONET/SDH to connect metropolitan and wide area networks together.

Further considering the delay, its responsive element arises from hidden parameters, congestion and buffer delay. Congestion and buffering delay can be probabilistic, when the routers are encumbered with random traffic load on their physical links. Congestion and buffering probability density functions (PDF) can be considered by Erlang traffic formulas [125], and Ethernet congestion delay is further explored in Figure 3.6.

3.2 Remote Airfield Communication Case Study

The safety critical context here is airfield communications, where SESAR [92] is progressing with the aim of producing a Europe-wide unified air traffic control infrastructure. SESAR proposes a management information model known as SWIM, which combines multiple traffic streams (concerning flights, weather and so on). The proliferation of connections inside this network opens up the issues of operation, maintenance and security. The provision of critical networking capability to each application offers enhanced prospects for network operation [18].

The case study concerning radar transmission between Glasgow and Coventry mentioned previously is further extended in its application using operational airfield radar data from multiple sites in Figure 3.1. The transmission of low payload but time critical radar information, transmitted at periodic time intervals (including blank transmission), could theoretically be achieved with a dedicated E1 link. The distance from Coventry to Glasgow is around 408 km and so there is a trip time delay of approximately 2 ms over copper cables. The E1 physical switching technology has a data bandwidth of 2,048 Mbps with 8 bit time slots, 32 time slots in a frame and 8000 frames/s [126]. Thus, the 136 bit (

Table 5-2) radar payload takes 17 time slots to complete and thus consumes a complete frame of 125 µs. Thus, when the radar transmits a packet every 5.7 ms, the total delay using E1 (radar plus protocol plus physical delay) is approximately 5.8 ms. This design requires a dedicated communication link which is expensive with low utilisation, no overheads making it incompatible with networking, and a low payload with fixed time intervals between transmissions.

Legacy radar equipment has a relatively low payload size requirement but a reasonably high message arrival rate (low time intervals between transmissions) and thus needs a high bandwidth to maintain its target of providing rapid client-server response. Investigation of current airport practice found that radar data packets had a high overhead and the effective data transmission efficiency was low (23.6%) [127] (Table 5-1) . Although a dedicated high bandwidth link from Coventry to Glasgow with the fastest response time could cover all overhead cost, maintaining such a service would be relatively expensive. One possible solution is to remove unnecessary overhead by packet to frame compression combining multiple packets into one, creating a more effective transmission, which is termed *IP defragmentation*. For time critical applications this is preferable to IP fragmentation [20, 128] which adds more overhead and produces a higher delay variance as a cost for lowering the time interval between transmissions [129].

3.2.1 Airfield Communication Network

The EU project SESAR entails the development of a net-centric information system, digital communication and layered adaptive security for ATM. SESAR plans to increase the size of the ATM network as well as the operation and management [3].

Each country in Europe has its own standards and approach to implementing a SWIM system, the core SWIM structure uses a host as an information management system for interoperation between different standards from other countries. This system gathers network information from a pool (raw information provided by the equipment suppliers) and presents it in segments [18]. The digital telecommunication ATM system in the UK uses TCP or UDP and this transport protocol standard was created by Euro-control [44]. In OSI model terms [7], the data is transported from the radar to the control tower either via copper wire or optical fibre as the Physical layer and Ethernet for the Data-Link layer, though these parameters limit traffic flow variation in the network. Assuming the system under test is unaffected by external factors (such as bit errors), a routine network maintenance protocol would send packets consecutively. When an external factor affects traffic flows, the novel technique devised in this research is used to recognise and identify the traffic flow, and discriminate any distortion in a network.

Airfield communications comprises of two types of link, one for air-to-ground and one for ground-to-ground telecommunications. This research investigates the characteristics of a radar communication link from air traffic radar to the control tower (ground-to-ground). An air-to-ground link would be subject to other delay factors such as radio distance, interference and signal to noise ratio, whilst a ground-to-ground link has fewer transmission problems but still suffers from network congestion problems. Although air traffic radar has been heavily regulated through standards regulation bodies such as Euro-Control's ASTERIX format, data gathered from an airfield has shown there are potential congestion delays within the network. The detailed radar communication traffic regulations can be found in the design for the level of quality assurance in industrial communication protocols [130]. The analytical work featuring

in the design of an intelligent communication traffic monitor uses SVM (Support Vector Machines) [119, 131, 132] to classify network congestion in a real time safety critical telecommunications network. Conventional techniques only inject additional regulatory protocols for network maintenance checks on a per application basis and periodically monitor the traffic conditions [133]. As the network expands traffic conditions become more varied, thus imposing additional network traffic protocol load is no longer an effective solution in managing and controlling network congestion, nor does it use the network bandwidth efficiently. Network traffic congestion is caused by the removal of network traffic shape regulation to embrace flexibility in custom applications. This intelligent traffic monitoring tackles unknown application and traffic conditions by using neural network techniques to catalogue, sample and identify trends where traffic congestion is occurring. SVMs are later used for recognising trends by supervised learning and using them to identify network traffic congestion.

3.3 Intelligent System for Optical Network Design

In this investigation, network traffic is collected and sampled though a novel time matrix, developed in this research. The method uses SVM, which is a machine learning technique that arose after the development of the artificial neural network (ANN) [134] of learning and optimising in Intelligent Systems Engineering (ISE) Methods [135]. In engineering, SVMs are known as pattern recognition and are used heavily in machine vision for industrial plants [136], whereas in computer science SVMs are referred to as part of machine learning . Pattern generation depends highly on the design of an experiment to capture the dataset. Thoughtful plans are required

to ensure all the detailed data characteristics are captured; all non-essential factors under observation are kept out of the design. Intelligent Systems research has two categories, technique optimisation and application example [117]. The former entails improving the execution speed for identification and classification, whilst the latter utilises SVM to discover extra knowledge within the field. Here the focus is on application research, where a choice may be made between using a training or a learning set. Both of these depend upon whether the pattern recognition system is supervised or not; training uses past results to identify patterns in output occurrences. A training set is normally created from a selection of captured results (supervised). The learning set uses inputs to recognise output responses (unsupervised) and heavily involves clustering (input compression). The pattern recognition stage uses a supervised training set to provide patterns to understand more about the network than previously known [137].

3.4 Data Analysis

Data was captured from Glasgow Airport (Figure 3.1) to provide insight into the IP packet based technologies they use for real time safety critical communications. The universal surveillance data format supported by all SESAR certified equipment is referred to as ASTERIX, a snapshot of which is shown in Figure 3.2 below. Although Euro-control continues to promote the ASTERIX format, which is similar to drafts for NATO STANAG 5535 Multi-version, the Glasgow Sensis equipment [1] follows an older standard created from Euro-control. The format is not recognisable by the Aircraft

[1] brought out by the Saab Group (15 August 2011)

Communications Addressing and Reporting System (ACARS) and the known ASTERIX format presents opportunities to develop a live data-link analyser for interrogation of network integrity.

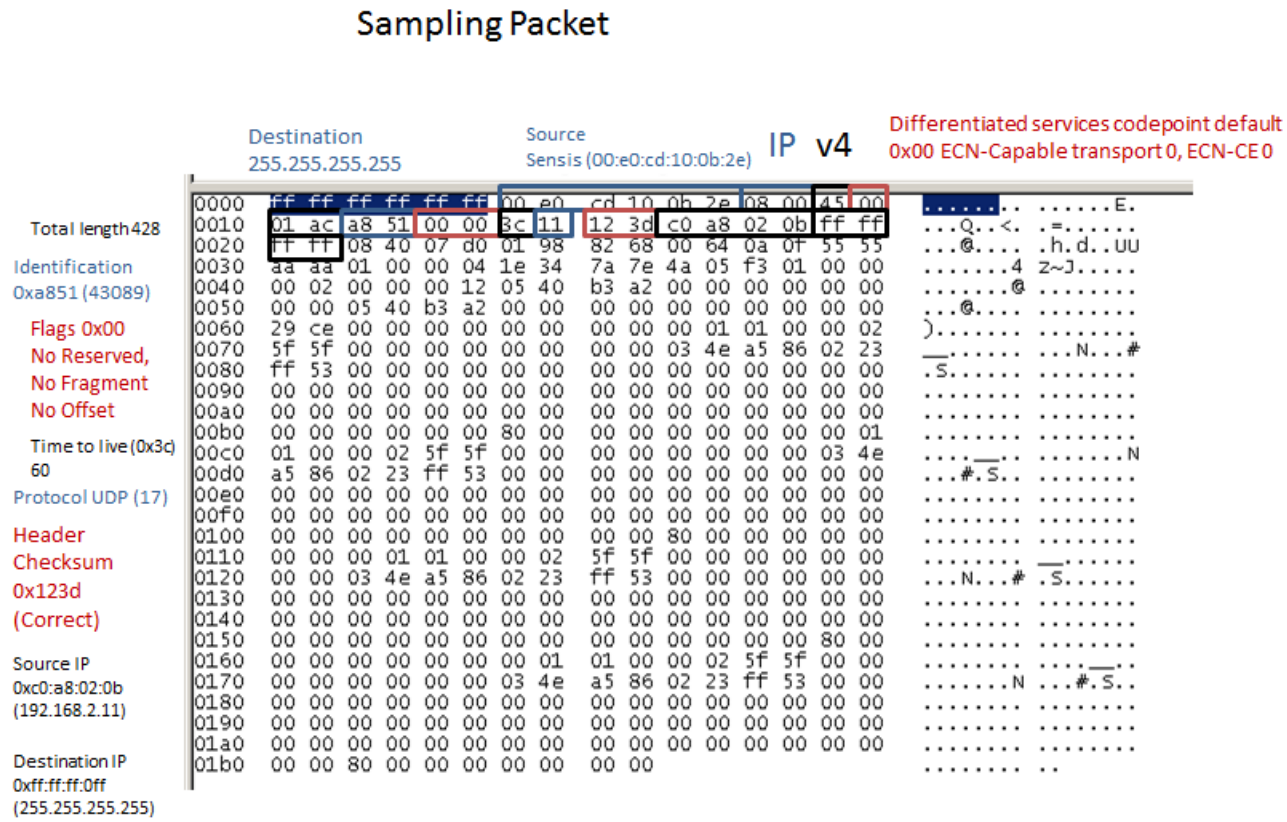


Figure 3.2 The breakdown of an Ethernet frame, following the standard of EUROCONTROL and SESAR-SWIM. Similarly all the protocols recorded in the network are identified

Using the network capture software Wireshark [178] it was possible to gain insight into and represent the captured data, allowing observations to be made. A packet is captured and analysed following the specification and structure explained in the OSI.

A breakdown of all the protocols existing in this airfield communication is shown in Figure 3.3.

This network does suffer from network congestion problems when it is busy (Figure 3.5). Conventional network monitoring techniques only observe the network from a statistical perspective [138] (where details are hidden and rounded up as arithmetic means), which causes real time data to be held inside the buffer of the proprietary multiplexer and increases time of transmission from a real time application. This congestion problem is caused by mismanagement of traffic by the external Cisco routers (in red), this is later recovered by a huge UDP packet spike.

This result is hidden from the network administrator using conventional traffic monitoring techniques Figure 3.4).

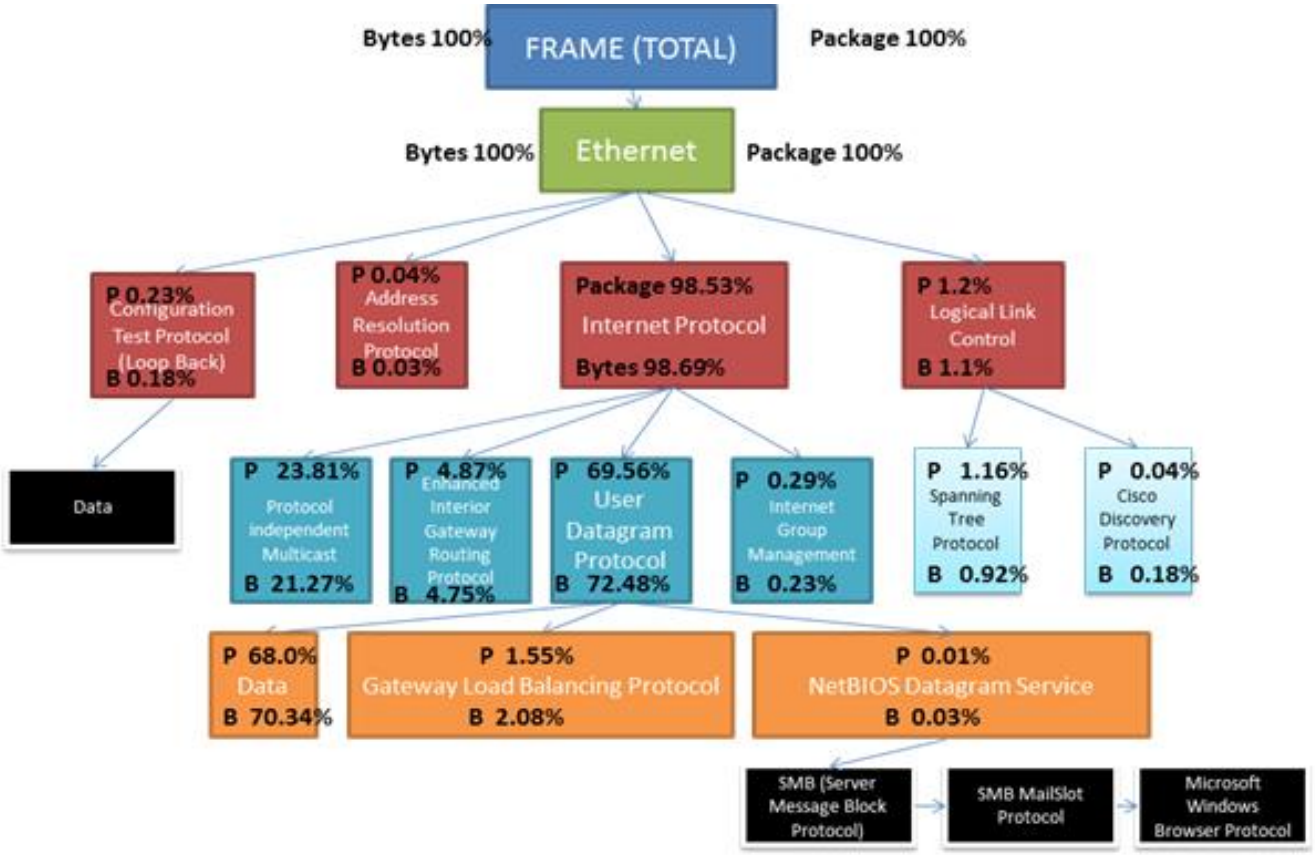
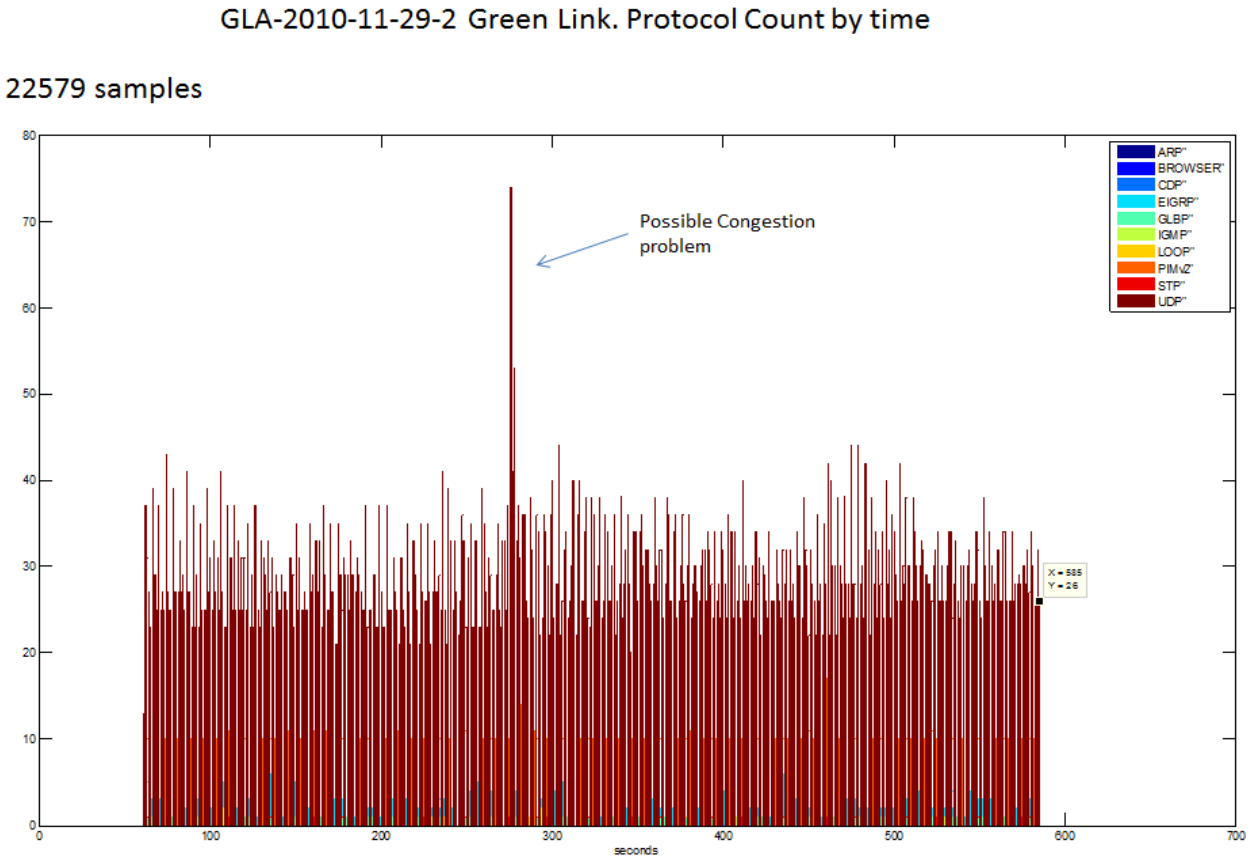


Figure 3.3 The captured data composite of Ethernet and other layer protocols. Some of the extra protocol data is found escorting the safety critical transmission (ASTERIX), this is labelled as data. P labels packets and B labels bytes

Figure 3.5 The number of packets against the time sample (seconds). Each of the packet is broken down into protocols; the UDP contains ASTERIX (radar data). The number of packet sparks from congestion over at the FTI multiplexers



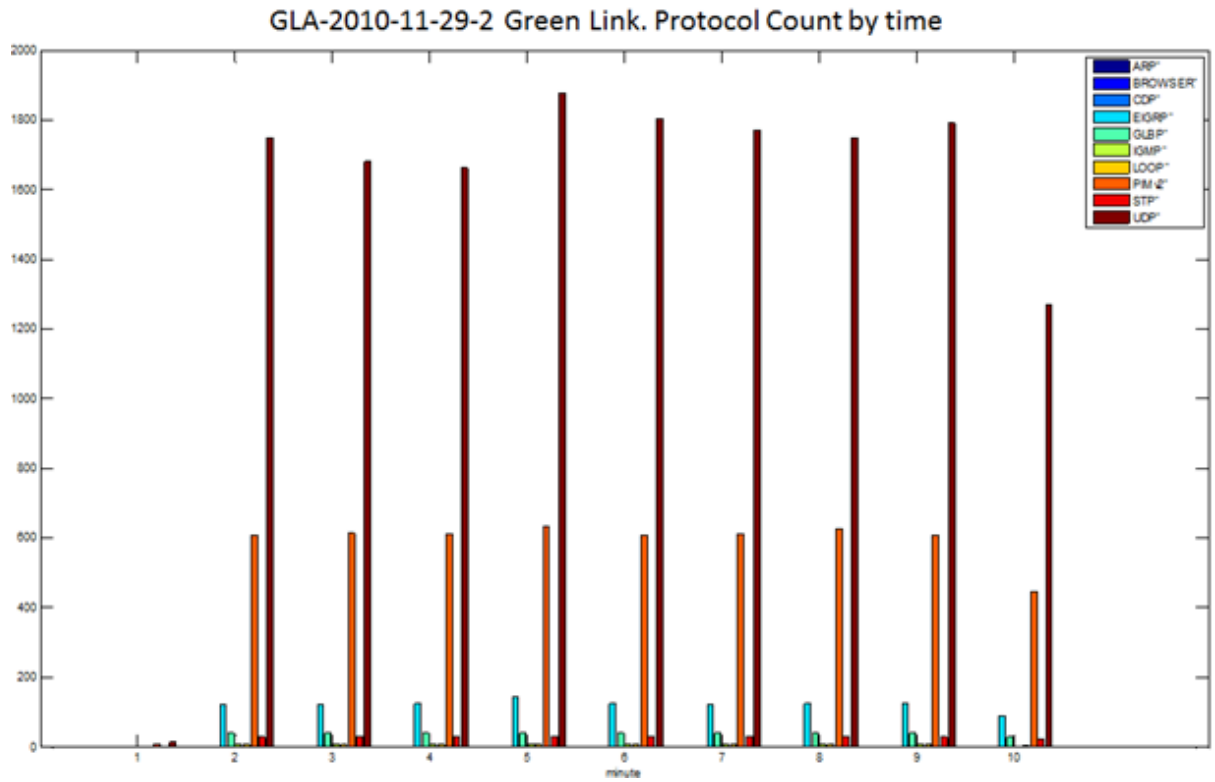


Figure 3.4 The same captured data from Figure 3.5 is represented in minutes (instead of seconds). The network congestion problem is hidden from the network operator as the same number of packets is received over each one minute interval

3.4.1 Window Size

Packet sequence numbers are randomly generated in a TCP transport protocol to keep track of packet transmission. This is a useful feature to keep tap of the number of window cycles per transmission, but eventually sequence numbers are reused. The time of reusing sequence numbers depends on the physical layer technologies, sequence numbers are 32 bits in length. In low physical data rate technologies, E1 with a 2Mbit/s data rate, the time after the sequence number is recycled is around 4.78 hours. A 32 bit Sequence Number generates 2^{32} number of packets, the lowest data payload is around 8 bits (2^3). A theory model of maximum data generation is around 34Gbit ($2^{32}x2^3=2^{35}$bit), the time to wrap around the same sequence number falls to the minimum data generated against the transmission rate. Problems occur when packets with the same sequence number appear in the network, though a general rule forbids packets to exist more than 60 seconds in a network. For technologies that have a wraparound time of less than 60 seconds (STM-4 and Gigabit Ethernet) you require sequence number extension [139] (Table 3-1).

Table 3-1 Maximum Time to wrap around sequence number based on bandwidth and trip timer

Data rate,	Maximum Time to wrap around
E1 2Mbit/s	4.78 hours
Ethernet 10Mbit/s	0.95 hours (57.27 minutes)
Ethernet 100Mbit/s	5.72 minutes
STM-1 155 Mbit/s	3.69 minutes
STM-4 622 Mbit/s	55.24 seconds
Gigabit Ethernet 1 Gbit/s	3.4 seconds

A sequence number extension is required for each individual physical technology to maintain effective communication. When a network suffers from 100ms round trip delay, the data rate is scaled as follows (Table 3-2).

Table 3-2 Reduce bandwidth based on trip timer delay

Data rate,	Size of window
E1 2 Mbit/s	$(2x10^6)$ x0.1=$2x10^5$ bit
Ethernet 10 Mbit/s	$(10x10^6)$ x0.1=$1x10^6$ bit
Ethernet 100 Mbit/s	$(100x10^6)$ x0.1=$10x10^6$ bit
STM-1 155 Mbit/s	$(155x10^6)$ x0.1=$15.5x10^6$ bit
STM-4 622 Mbit/s	$(622x10^6)$ x0.1=$62.2x10^6$ bit
Gigabit Ethernet 1 Gbit/s	$(1x10^9)$x0.1=$100x10^6$ bit

3.5 Methodology

Using the Wireshark software and a laptop, network traffic data was recorded to a file. This captured packet data comprises of the type of protocols in use, payload size, and transmission time. Conventional monitoring techniques, such as statistical averages, are used for identification, sorting and focusing the level of detail between analysing and sampling network traffic in real time. A captured packet file can be expanded to a list of chronological events about packets arriving at the investigation point in a network; this information is not processed in real time but stored as historical records for discovering network faults. This method neither offers the capacity for detecting performance issues in real time, nor provides an accessible method of detection to

localize and diagnose performance issues. A network traffic analyst responds to a fault in a network before investigating the captured packet file to troubleshoot a network. This research presents a method to provide insightful real time network traffic analysis, to detect where problems occur in a network.

3.5.1 Problem Statement

Each application has specified transmission characteristics that are hidden when recorded from a network link perspective. This task was made difficult because of UDP radar transmissions having no sequence number. The unknown hidden contents are payload size and specific time sequence duration. For example, a ten payload per second application could transmit a series of payloads in that one second; it can be one payload for every one tenth of a second, or no payload in ninth tenth of a second, but the whole payload (ten) in the last one tenth of a second. This sequence is in-built by the design of the radar equipment communication manufacturer, but never specified by the regulatory standard of the safety critical network operational management. Without this prior-knowledge of the equipment transmission sequence design, it is difficult to determine and observe whether any congestion delay has occurred in those incidents. The first challenge is to discover the transmission sequence of the radar application, and using this control transmission sequence to observe and analyse whether there are network congestion delays amongst the various communication equipment in the network. The first step undertaken in the investigation is to create variable time sampling windows, populated by the payload of the application. A high resolution time sampling window does not necessary reveal the

transmission sequence [140]. This method is rather an iterative sampling process for focusing and determines a reference point of payloads per second to create a transparent application transmission sequence for observation.

3.5.2 Network Traffic Transmission Sequence Monitoring

Every transmission sequence is recorded using a packet capture device and this device operates at the top layer of the OS, these transmission sequences can be separated by several factors; the protocols in use, MAC addresses, and network addresses. This packet information reduces the complexity of the analysis by isolating transmission sequences into different sampling matrices by these categories. A reduction in varying payload rate of change improves the clarity of the traffic congestion observation; a focus sampling matrix allows the observation of inter-dependent protocols operational relationship and its cause and effect, and offers performance management such as providing the exact amount of bandwidth for these real time protocol requirements. The general characteristic transmission sequence can further be reduced to binary level observations, either identified as known or unknown. A detailed level of observation computes the payload difference to spot positive and negative transmission progression sequences (examples such as arithmetic or geometric sequences), a combination of both type of sequences is known as a chaotic sequence [141]. Analysing the structure of each transmission sequence provides insights to adjusting matrix perspective and detecting anomalous patterns in transmissions.

3.5.3 Transmission Sequence.

A payload can experience three states, stationary, transmitting and accelerated transmission. In the stationary state, the payload is simply being stored inside a buffer, waiting to be transmitted, this occurs when a network experiences congestion. When this occurs it triggers a reaction response from the transmitter for an increase in payload traffic to compensate the drop in communication due to buffer retention. Only the payload and time are recorded by the traffic monitor at the point of a network switch. This creates a density estimation problem of discovering the hidden variable rate of payload per second transmission rate and its acceleration during network congestion. The solution can be found by understanding the operation of the application in terms of its geometric transmission progression from packet payload per time slots. The acceleration of radar payload traffic increase is only temporary, identifying and analysing the network condition requires the construction of a focused sampling matrix, this allows a coordinated system of identifying reoccurring network congestion condition in the network by discrete time sample analysis. Figure 3.6 demonstrates network congestion with a known deterministic payload increase from identifying the operation of the radar transmission. However, when the upper two states are hidden (payload per second and acceleration), these spikes are often difficult to analyse as their duration is temporary and the payload size is random. This randomness is caused by other application traffic demands and the limitation of link capacity from the radar to the control tower. The unknown application and its payload transmission add more layers of uncertainty regarding the correct operation of the network. An application may only be designed to transmit twelve payloads per time slot, but as soon as there is network congestion, the critical time element of this

application triggers an increase in traffic, which further increases the communication delay. The challenges are these application requirements just by observing the number of time sample and the payload in time slots.

Time Sample	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Payload	0	6	24	54	96	150	216	294	384	486	600	726	864	1014	1176	1350	1536	1734	1944	2166
Payload/sec	0	6	18	30	42	54	66	78	90	102	114	126	138	150	162	174	186	198	210	222
Payload/sec²		0	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12

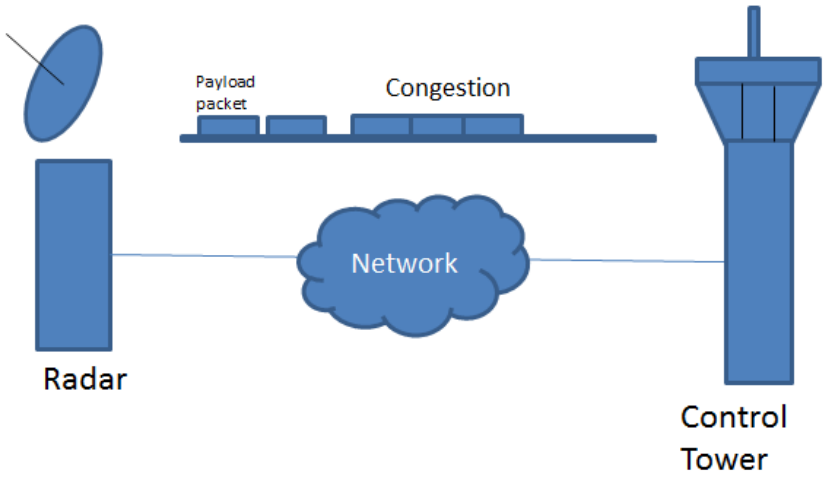


Figure 3.6 The rate of change in payload by the three layers perspective of communication and networking

3.5.4 Basic Traffic Time Matrix Construction

The imperative feature in this design is that there is no physical packet buffer introduced to sample and collect traffic in the network, as this would skew the transmission pattern discovery and hinder any meaningful sequence identification. A physical unmanaged packet buffer causes congestion delays in a time critical network. Packets are recorded and copied in a virtual buffer, the size of which is dictated by the processor and memory in the device used. A sampling matrix is constructed in a square matrix containing a number of payloads stored in a time slot transmission

sequence; the sampling matrix coordinate is two dimensional (t_{xy}), and is taken from discrete time events recorded from the virtual buffer. The sampling matrix is synchronised and time stamped from the virtual buffer time instances. The sampling matrix is populated by packet instances (*n*) and stored inside the time slots as discrete time instances (t), these time instances are chronologically ordered (Figure 3.7a). Transmission sequences are folded on top of each other in a square matrix (Figure 3.7b). Originally, the square matrix is fixed in size, but the size is dynamically adjusted according to the sequence. Each consecutive packet instance appears at the start of a matrix (bottom left), and the last instance is pushed away at the end of a matrix (top right). When there are two or more consecutive blank time slots (zero packet transmission), the sampling matrix discards blanks in the square matrix. The pattern recognition process operates after the sampling matrix format is filled with packet instances.

3.5.5 Sampling Matrix Format

Packets vary in payload, but will not exceed the Ethernet frame data size, which is fixed by the Ethernet standard authority [42]. A transmission stream of packets is comprised of variable time intervals between packets, and variable payload in that one packet duration incident. The absolute theoretical maximum limit to these variations is that the whole recorded dataset only contains two packets, thus the whole dataset contains one long time interval. The smallest measurement of duration in this data transmission stream is the physical limitation of time event detections from the packet capture device, and the minimum acceptable payload size of an Ethernet frame. This

method simplifies discrete time events as time slots, and number of packet instances as discrete units. Each passing time interval with no packet is a blank time slot in the transmission sequence. Each time instances records the number of packets in that particular time slot. A simple packet transmission sequence is a list of natural numbers connecting the number of payload for that particular application protocol, an advanced study of revealing hidden transmission sequence is carried out by the technique below.

$$M_{series} = \prod_{x,y,}^{n} det\left(l_{xn,yn}\right)_n$$

Equation 3-5

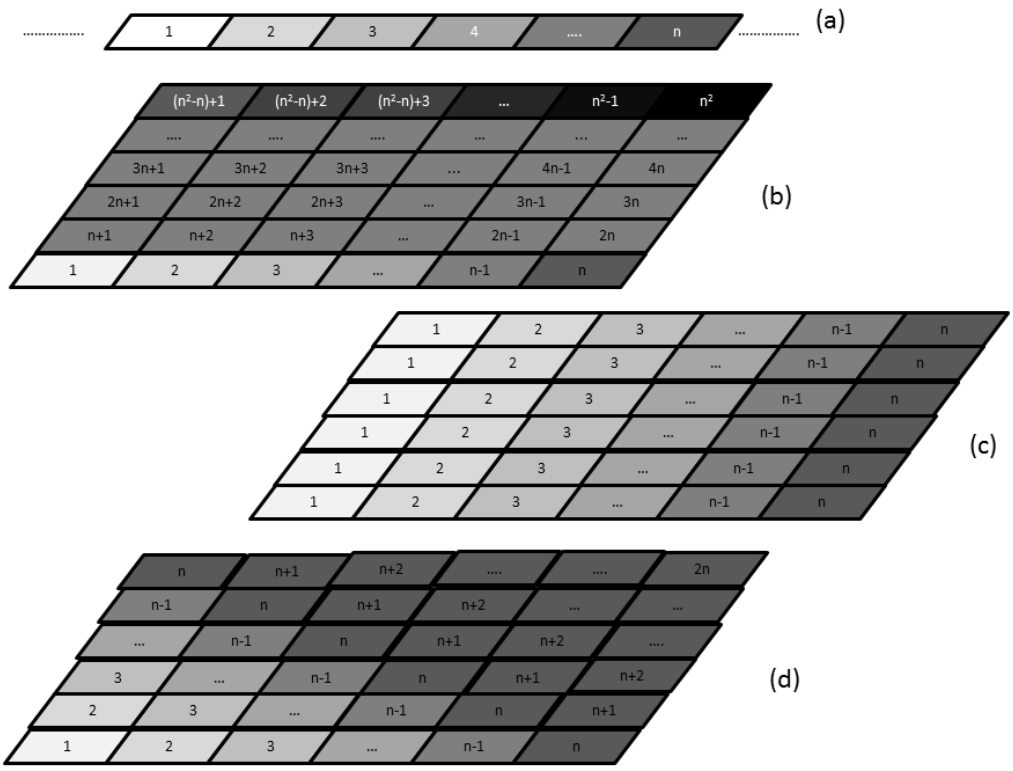
$$L_2 = m * x = \begin{bmatrix} m_1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ m_2 & m_1 & 0 & & 0 & 0 & 0 \\ m_3 & m_2 & m_1 & & 0 & 0 & 0 \\ m_4 & m_3 & m_2 & & & & \\ m_5 & m_4 & m_3 & \ddots & & \vdots & \\ m_6 & m_5 & m_4 & & & & \\ & \vdots & & & m_{n-3} & m_{n-4} & m_{n-5} \\ & & & & m_{n-2} & m_{n-3} & m_{n-4} \\ 0 & 0 & 0 & & m_{n-1} & m_{n-2} & m_{n-3} \\ 0 & 0 & 0 & \cdots & m_n & m_{n-1} & m_{n-2} \\ 0 & 0 & 0 & & 0 & m_n & m_{n-1} \\ & & & & 0 & 0 & m_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$

Equation 3-6

$$L_3{}^T$$

$$= [m_1 \quad m_2 \quad m_3 \quad \dots \quad m_n] \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & & & & 0 & 0 & 0 & & \\ 0 & x_1 & x_2 & x_3 & x_4 & x_5 & \cdots & & & 0 & 0 & 0 & & \\ 0 & 0 & x_1 & x_2 & x_3 & x_4 & & & & 0 & 0 & 0 & & \\ & & & \vdots & & & \ddots & & & & \vdots & & & \\ & & 0 & 0 & 0 & & & x_{n-3} & x_{n-2} & x_{n-1} & x_n & 0 & 0 \\ & & 0 & 0 & 0 & & \dots & x_{n-4} & x_{n-3} & x_{n-2} & x_{n-1} & x_n & 0 \\ & & 0 & 0 & 0 & & & x_{n-5} & x_{n-4} & x_{n-3} & x_{n-2} & x_{n-1} & x_n \end{bmatrix}$$

Equation 3-7



a) Dataset extrapolation, packets are arrange into a sequence in time, (n) is the magnitude (the number of packet payload) in the system.

b) Packet sequences are arranged in a square matrix (folding) for spotting pattern within sequences. Sequences with the same magnitude are line up to determine the periodicity.

c) The level one square matrix is resized and shifted until periodic sequence emerges. The periodic sequence is used as a matched filter for remove repeat pattern in the analysis.

d) The level two square matrix is used as an inference matrix to cross-correlate, the start and the end of new and old sequences.

Figure 3.7 The design of an intelligent sampling matrix

3.5.6 Pattern Analysis

Pattern analysis in network traffic is achieved by determinant calculations (Equation 3-5) in order to find the sum of the payload series expansion in the sampling matrix. The determinant calculation indicates the rate of change between the matrix coefficient and its three adjacent neighbours in the sampling matrix. A two by two determinant matrix is initially used to discover the inter-relationships between the closest matrix coefficients (between payload and packet). The determinant matrix calculation will slide onto the sampling matrix indefinitely until the last two by two elements of the matrix are matched (Equation 3-6). A number series (*m*) is produced from determinant calculations for identifying the transmission sequence to determine the rate of change to conclude which three states the payloads are in. The *m*-series contains all the matrix element coefficient inter-relationships in a sampling matrix (Equation 3-7). When the multitudes of discovered *m*-series are cross-correlated, the determinant matrix product will be zero; the group of element coefficients are completely synchronized. A longer zero *m*-series chain means a longer matching transmission sequence, and subsequently, the more transparent the periodic transmission sequence becomes. The group is then registered as the first pattern transmission sequence. A sampling matrix with a complete zero *m*-series chain represents total matching of all the transmission sequences in the sampling matrix (the net payload per second is zero). The value in the *m*-series describes the perspectives of both the sampling matrix and the transmission sequence.

3.5.7 Matrix Perspective Analysis

The sampling matrix and payload sizes can be explained from three perspectives: variation zero (stationary payload), variation one (payload/s) and variation two (payload/s^2). The variation zero sampling matrix is a square matrix containing an overlay of packet instances that represent the total payload instances against the highest sampling of time incidents (stationary cumulative payload). The variation one matrix contains the determination calculation of the group payload incidents measuring the past and future rate of change in payload per second. Its matrix coefficients are replaced with the determinant values of the variation zero matrix (*m*-series); the variation one matrix is known as the first order payload against time sample. This process is repeated for the variation one *m*-series matrix on to the second order variation two matrix measuring payload/s acceleration. All packet instances have many degrees of freedom from the direction of payload externally triggered from network equipment operation, through the protocol design to the design specification of the application. Moreover, it is important to build up a stable payload platform map for distinguishing the normal behaviour of the payload transmission from an anomalous event, namely packet network congestion, which disturbs the historical sequencing of transmission. A focused sampling matrix has the same number of rows and columns as the number of instances in the normal transmission sequence of the network. When the sampling matrix row is mismatched with the transmission sequence, the sampling matrix (Figure 3.7d) has a mismatch of one packet instance less than the one in the transmission sequence, the *m*-series are no longer measuring the magnitude of the payload difference in time (payload per second), but also measuring the direction of an unfocused sampling matrix (Eigenvectors). This diagonal line (±45

degree) points to whether the sampling matrix row has one more or one less packet instance than the transmission sequence, the direction of the triangulated Eigenvector direction points toward the calibration requirement of focusing the sampling matrix (Figure 3.7c). A matched sampling matrix (Figure 3.7b) would be at ±90 degrees packet instance to packet instance coordination. This diagonal line angle changes depending upon the distance separation between the two points (the same packet instances measured from t_1 or t_n). This method allows a depth perception to increase or decrease the size of the sampling matrix (rows and columns), by focusing the sampling matrix within rate of change (*m*) up to the same number of time sample (t_n) of the magnitude. The distance between key packet instances (Eigenvector calculation), is the fundamental concept to expand the size of the sampling matrix, to encompass all degrees of rate of change (n^2) in packet instances. Hence, a completely random packet instance sampling matrix with a maximum of multiple (n^2) degrees of movement packet instances has no pattern.

Pattern analysing these transmission sequences provides additional knowledge about the operation of each packet transmission category. For instance, a binary transmission sequence is often seen between network management devices. An example is network time protocol, which has fixed time slot intervals. These transmission sequences are matched instantly by a binary *m*-series, showing the end of a transmission sequence (skipping all double blank time slots). Positive *m*-series indicates variation one and negative one *m*-series indicates variation two sampling matrix. A Change in transmission sequence indicates early problems in network devices. This is an alternative method to rapidly match a sampling matrix (variation zero), by the power of *m*-series and transmission pattern sequence.

Two number transmission sequences have only two unique numbers of packets (two rate of change *m*-series) and zeroes, There are only four unique *m*-series and zeroes. The body of the transmission sequence is marked by zeroes and non-zero *m*-series mark where sequences are mismatched (start and end); the end of a transmission sequence is indicated by the four unique *m*-series. This is used to identify the mismatch start and end of any two transmission sequences. A direct example of protocol pattern recognition process at work is a Cisco Discovery Protocol (CDP), this pattern recognition matrix only requires the position to be shifted in the column to show the periodic transmission sequence (Figure 3.8).

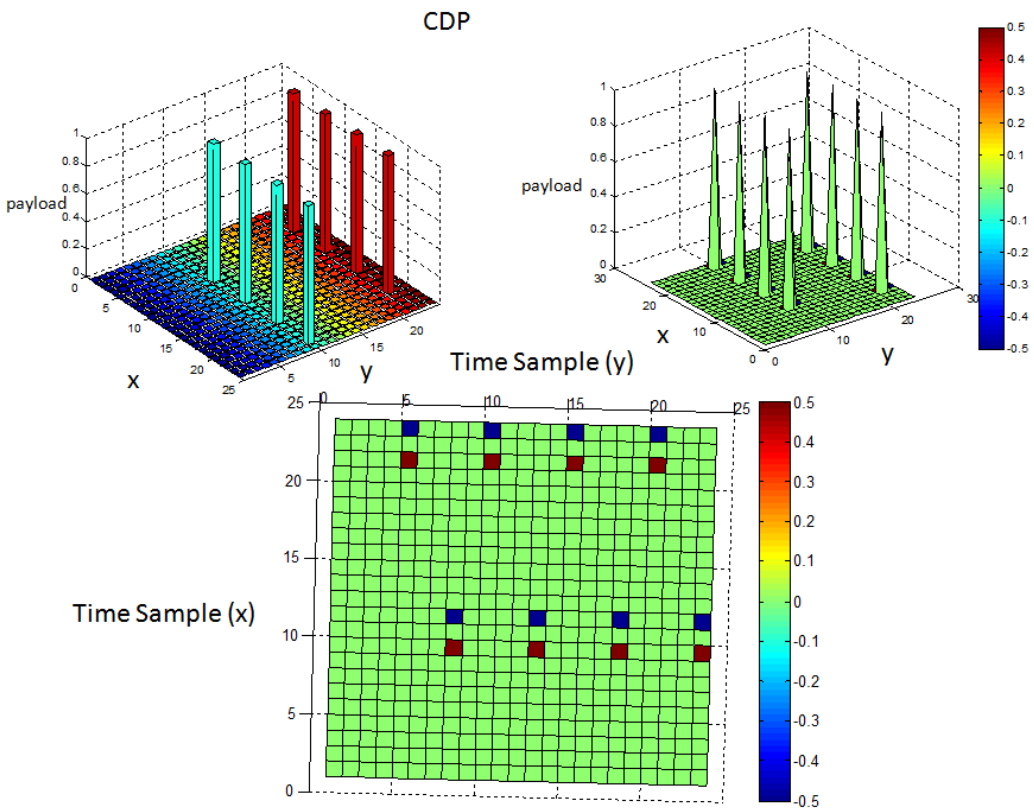


Figure 3.8 The appearance of patterns in the data from the Cisco Discovery Protocol (CDP)

3.6 Binary Radar Congestion Identifier

Binary radar transmission possesses two parameters, a fixed packet payload identifier and a fixed time interval between transmissions. If the transmission model is certain, the transmission exhibits periodic qualities, such as a step function. When there are parameter variations, these may be thought of as the probabilistic properties of the dataset for both parameters. Furthermore, it is possible to consider that there are two additional parameters in the dataset, the properties of the payload affecting the time interval and vice versa. These are often hidden if one only analyses the payload or time interval variations. In packet communication, there are uncertainties. A one-dimensional statistic can characterise a dataset by its arithmetic, geometric or harmonic means. A two dimensional one can characterise the spread of variation, central tendency and moment generation by distribution. A maintenance protocol is simulated in a point-to-point network and has a statistical property of a discrete uniform distribution for both payload and time interval (Equation 3-8). This is rewritten using the Heaviside step function (Equation 3-9). The combined statistical distribution has a relationship that the probability of occurrence is the probability of payload and the probability of time occurrence.

$$P_{(x,t)} = \begin{cases} \dfrac{1}{b-a} & a \leq x \leq b \\ 0 & otherwise \end{cases}$$

Equation 3-8

$$P_{(x,t)} = \frac{H(x-a) - H(x-b)}{b-a}$$

Equation 3-9

3.7 Density Estimation and Pattern Recognition

The two parameters, packet payload and the time between transmissions, were simulated using a discrete uniform distribution. The arithmetic mean of the payload around 70 kilobytes (double rate) and the time interval was ten seconds. Both the payload and transmission intervals parameters were combined (Equation 3-10). A time sampling window was created using three times the arithmetic mean (μ) of the time interval (Equation 3-11), sampling at 1Hz. The two pattern recognition window was constructed using the two greatest common factors of the one and a half of the arithmetic mean (Equation 3-12) (Equation 3-13).

$$y\left(P_{(x)}, P_{(t)}\right) = P_{(t_1)}P_{(x_1)} + P_{(t_2)}P_{(x_2)} + \cdots + P_{(t_N)}P_{(x_N)} = \sum_{N=1}^{N \leq (2\mu)^2} P_{(t_N)}P_{(x_N)}$$

Equation 3-10

$$\sum_{i=1}^{i \leq 2\mu} \sum_{j=1}^{j \leq 2\mu} P_{(t_{i,j})}P_{(x_N)} = \mathrm{W}\begin{bmatrix} P_{(t_{1,1})}P_{(x_1)} & \cdots & P_{(t_{1,30})}P_{(x_{30})} \\ \vdots & \ddots & \vdots \\ P_{(t_{30,1})}P_{(x_{871})} & \cdots & P_{(t_{30,30})}P_{(x_{900})} \end{bmatrix}$$

Equation 3-11

$$\sum_{i=1}^{i \leq 3} \sum_{j=1}^{j \leq 3} A_{(t_{i,j})} \begin{bmatrix} P_{(t_{1,1})}P_{(x_1)} & \cdots & P_{(t_{1,3})}P_{(x_3)} \\ \vdots & \ddots & \vdots \\ P_{(t_{3,1})}P_{(x_7)} & \cdots & P_{(t_{3,3})}P_{(x_9)} \end{bmatrix}$$

Equation 3-12

$$\sum_{i=1}^{i\leq 5}\sum_{j=1}^{j\leq 5} B_{(t_{i,j})}\begin{bmatrix} P_{(t_{1,1})}P_{(x_1)} & \cdots & P_{(t_{1,5})}P_{(x_5)} \\ \vdots & \ddots & \vdots \\ P_{(t_{5,1})}P_{(x_{21})} & \cdots & P_{(t_{5,5})}P_{(x_{25})} \end{bmatrix}$$

Equation 3-13

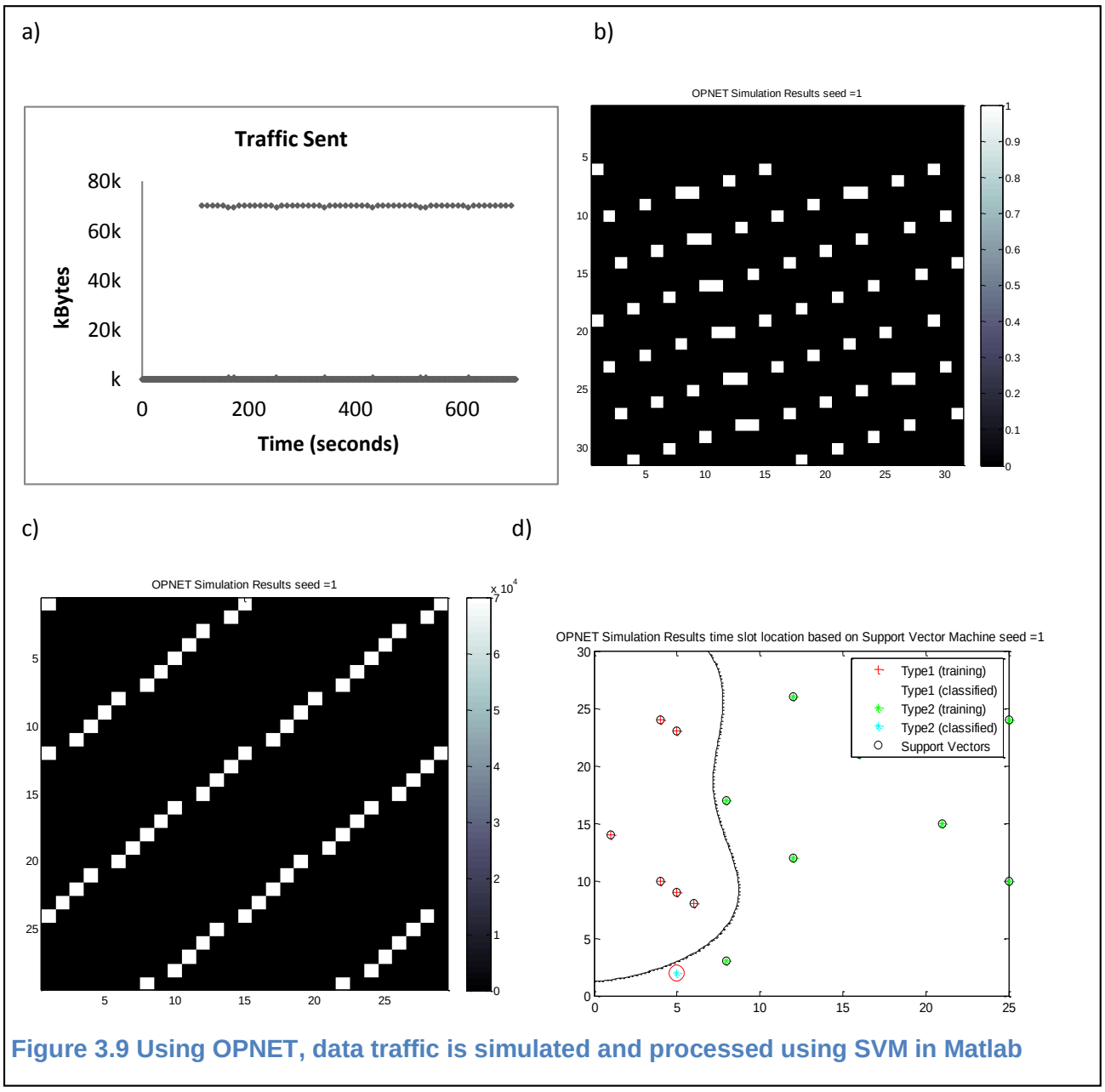


Figure 3.9 Using OPNET, data traffic is simulated and processed using SVM in Matlab

3.8 Results and Discussions

700 sample points were taken from the Radar traffic by optimising the payload/s parameter in Figure 3.9a. The original pattern recognition window used 31 by 31 to yield 961 samples (Figure 3.9b). A third parameter was discovered as the starting transmission offsets. The time sample offsets (variation in traffic starts) were removed (Figure 3.9c) and this pushed the data sample point forward in the windows (30 by 30). This matrix showed a more synchronized pattern than that without offset removal. Using SVM, two patterns were identified. There were two training sets, type 1 contained a smaller grouping of traffic payload pattern than type 2 (Figure 3.9d). The construction of the two pattern matrices for types 1 and 2 was via equations (Equation 3-12 and Equation 3-13), with a diagonal pattern on both. Matrix B (type 2) was used first to identify any large payload within its window size by sliding its window across the overall 30 by 30 Matrix, with Matrix A (type 1) subsequently utilised to identify potential congestion delay. The line is generated from the SVM polynomial non-linear classifier (kernel trick) [12] and represents the region where type 1 payload clusters split from type 2. This region line model was generated from one sample seed of the simulation only.

3.8.1 Payload Transmission Sequence Density Approximation

An arithmetic progression sequence has only one layer of progression. The same *m*-series value is produced when two of the same arithmetic progression sequences slide on top of each other (exception when matching occurs); the largest *m*-series value

indicates the end of a mismatch sequence (similar to two number transmission sequences). For instance, an arithmetic progression sequence has a progression of even numbers (e.g. 2, 4, 6, 8), exactly one mismatched time slot yields *m*-series values of four (positive or negative), exactly two mismatched time slot has *m*-series values of eight, three mismatched time slot has *m*-series values of twelve and so forth. In general, the arithmetic progression starts at a fixed n^{th} value, with a progression of the difference between the two numbers (d). When the pattern is mismatched in a sampling matrix, determinant calculations of *m*-series are the square of the arithmetic progression (d^2), creating an additional layer known as a geometric expansion series.

The largest *m*-series marks the furthest distance away from two matching sequences (similar to all previous progression sequences). This is same for arithmetic geometric sequences. Natural number sequences are completely unknown, depending on their position and rate of change (*m*). They can generate the same number of unique *m*-series as the rate of change, when two natural number sequences slide on top of each other. There is no shortcut method with these sequences apart from having prior knowledge about the expansion progression sequence. This knowledge can be obtained by statistical learning and combines observing repeating sequences and *m*-series; any repeating number sequence will eventually be normalised into a zeroes *m*-series matrix (variation matrices). A negative progression sequence becomes a positive progression sequence when the sampling matrix is inverted.

In retrospect, a recognized transmission pattern starts when the sampling matrices (Figure 3.7b) match the variation zero sampling matrix (Figure 3.7c). These sequences are recorded and set as a registered known transmission; the deformation of these pattern transmission sequences indicates problems in a network. This method provides instant data for a network analyst to spot other irregular transmissions in the

network, and remove known transmission sequences from unknown ones. A number sequence circuit cross examines transmission pattern from sequences, this either reinforces that the new sequence is just a small variation of the first sequence or highlights changes in the network. Stable periodic binary and two number transmission sequences (common in a network) are summed into an arithmetic progression sequence (sequence compression), this ideal sequence is valuable for sampling matrix clarification and to swiftly cross examine a network for the task above. The final products are patterns that help to determine whether there is packet congestion, packet dropout, or cyber-attacks. Packet transmission streams are similar to streams of rain drops showering an invisible man - the metaphor for hidden condition in the network affecting communication streams. When this method cross references a small area (the three variation of sampling matrices), the distortion in the rain drop (transmission sequence) reveals the appearance of the invisible man (fault in a network).

3.8.2 Hidden Properties in Data Analysis

The next step is to process the *m*-series matrix so as to identify the hidden inter-relationship between payload and packet. ASTERIX Radar communication has been determined as having a fixed payload transmission progression using the above technique on ASTERIX data. The result is a progression of a six payload increase rate (congestion) in the *x* direction and twelve payloads in the *y* direction. A focus time sampling matrix places this characteristic in focus form, even though on the surface the packets/s rate fluctuates constantly. This discovery can be demonstrated by

identifying the correct perspective of this hidden principle. The first step is to calculate the determinant value using the multi-level sampling described earlier. The radar transmits packets over the two links in the airfield, these transmission progressions can be caused by the safety critical link delivering a pattern of six packets in real time and the other link delivering a pattern of twelve packets respectively. A complete cycle of 90 time slots was taken in the focused sampling matrix. The Langrage multiplier is used to identify the maximum likelihood that the radar exhibits a constant payload per second, by the convergence of the two transmission patterns. The final estimate is shown in Figure 3.10.

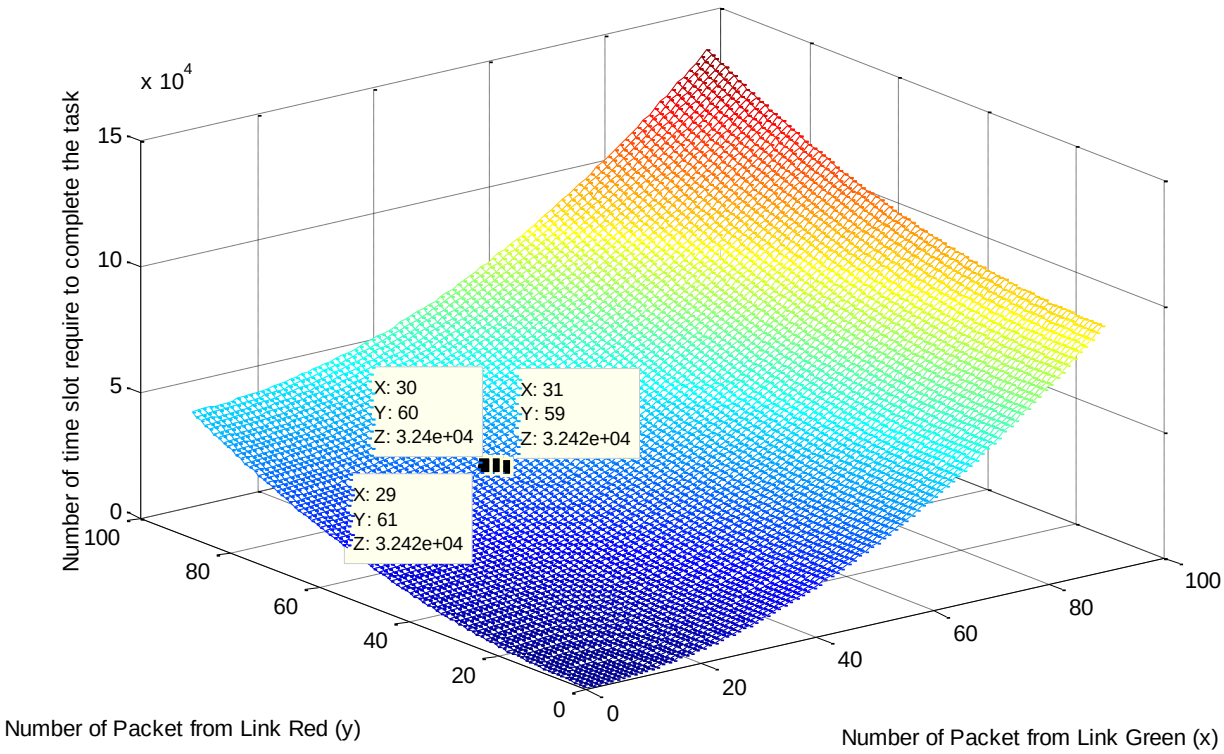


Figure 3.10 Lagrange multiplier solution of the minimum payload packet that is common between the two links and estimating the correct payload per second perspective for pattern recognition

The analytical expressions can be worked out by hand and are as follows:

$$t(x,y) = 6x^2 + 12y^2$$

$$t(x,y,\lambda) = 6x^2 + 12y^2 - \lambda(x + y - 90) = 0$$

$$\frac{\partial t}{\partial \lambda} \rightarrow \lambda = 720; \rightarrow x = \frac{\lambda}{12} = 60; \frac{\partial t}{\partial y} \rightarrow y = \frac{\lambda}{24} = 30$$

$$\Rightarrow t(60,30) = 32400 \text{ bits}$$

Equation 3-14

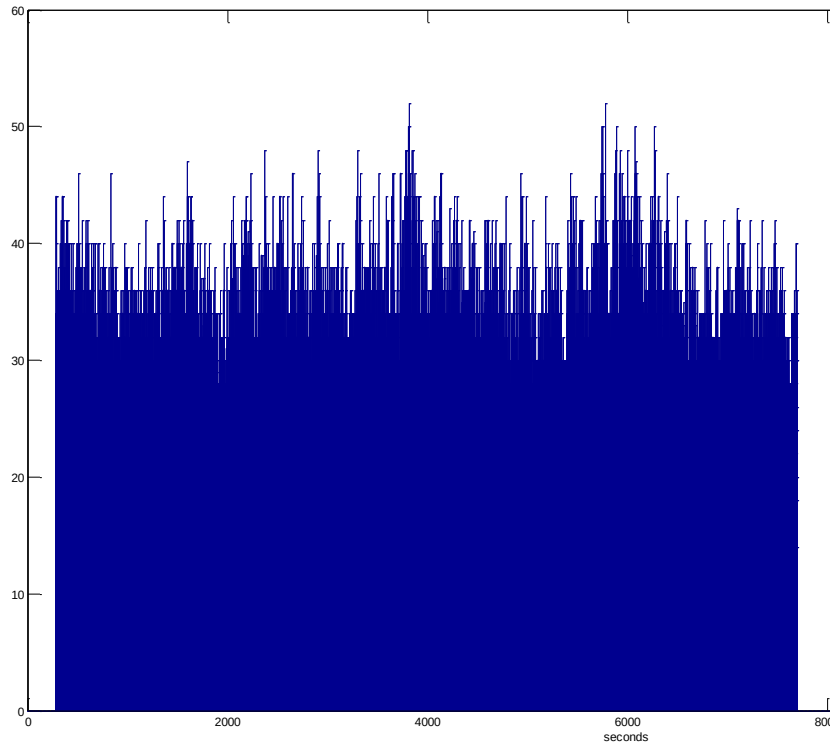


Figure 3.11 Packets per second recorded on a live Radar communication

The next step is to reshape and adjust the time sample window using the mounting point of the ~32 kbit transmission rate as the measuring point of the radar communication transmission sequence. The untreated packets per second in Figure 3.11 (by assuming each packet as one payload) can create a transmission

characteristic that looks random, only by combining the progression sequence of 10 samples (10 seconds) does the transmission periodicity become more transparent (Figure 3.12); the disturbed increase in the transmission sequence indicates a potential network congestion problem.

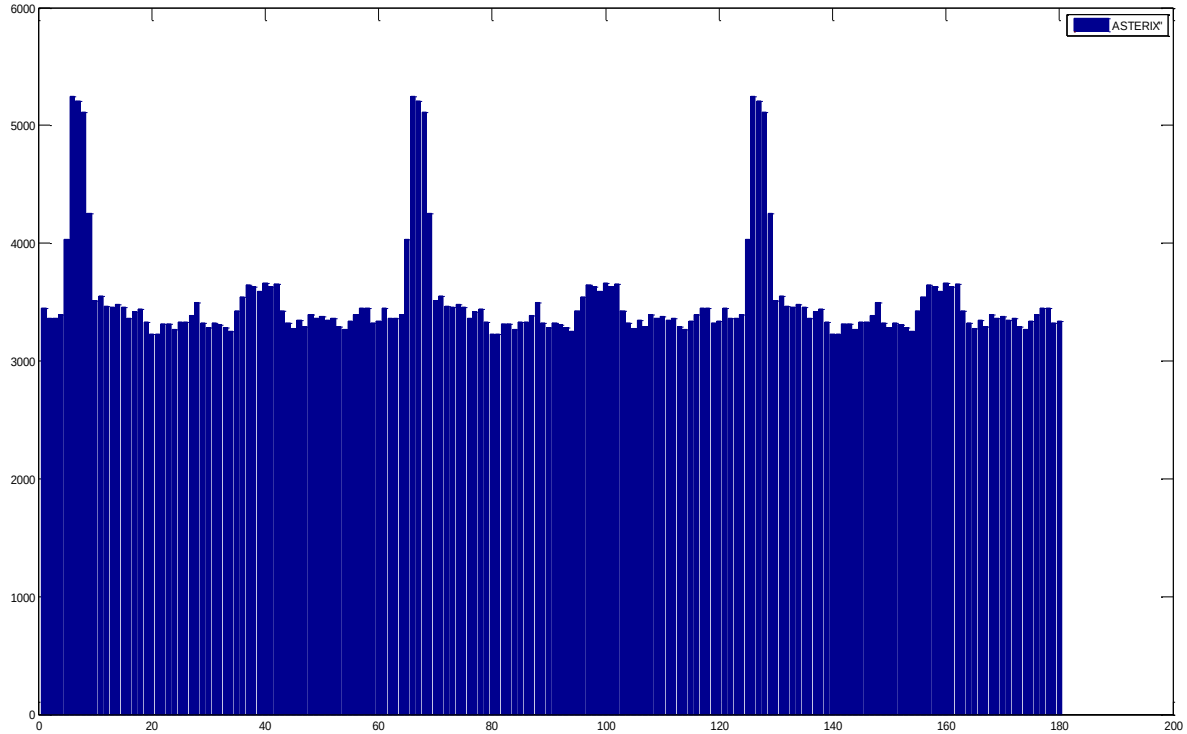


Figure 3.12 Payload (bits) against the correct time sample (10 seconds), in payload per second, of the live radar communication

3.9 Real-time Pattern Recognition for Congestion Detection

This packet inspection tool generates captured traffic data streams; real time data stream analysis is difficult. This section demonstrates the power of real time packet monitoring in a time critical communication network. Pattern recognition is the ultimate

solution to reduce processing power whilst maintaining the highest clarity of packet inspection. This method comprises the following stages:

(i) the construction of an insightful sampling matrix to encompass patterns in packet transmission;

(ii) the dissection of transmission patterns formed within transmission sequences;

(iii) removing all known transmission sequences from unknown ones.

These tasks are not readily available; the first task is to divide a single traffic data stream into multiple data streams by known network categories. It is possible for a physical network device to have several network ports (MAC addresses), several IP addresses (network routing addresses), and also to run several network communication services (protocols) concurrently. This inquiry partitions a single traffic data stream into these fundamental network components. These network transmission components have the lowest order transmission sequences (binary or two number sequences). These components transmit differently depending on the network configuration, network topology, number of senders, type of application and the configuration of the router algorithm setting. These low order transmission sequences become high order when they combine into one network category (e.g. IP addresses only). This analysis benefits from selecting periodic low order transmission sequences and compressing these into an arithmetic transmission sequence. When a known arithmetic transmission sequence becomes a high order incoherent transmission sequence, this provides an instant clarification when unfamiliar transmission occurs.

Methodology constraints are imperative in this design. The transmission sequence rate of change (*m*) cannot exceed the dimension length of the sampling matrix. This study proceeds by either expanding the matrix to encompass all degrees of freedom, or separating one study into sections. Sections only apply a small set of degrees; a high variation rate of change (*m*) matrix is dissected into two or more sampling matrices. A matched filter (a logic number sequence) is used to remove known patterns from unknown ones in the sampling matrix. One variation and two sampling matrices offer a confidence check between discovering the start and the stop of a transmission pattern by determinant calculations. These matrices act as an intermediate stage, for adapting new trends in the transmission sequence and to check for consistency in pattern recognition.

By way of example, a toy model considers a packet transmission system that sends an extra packet at every indicator interval (arithmetic transmission sequence), this resets for every ten packets. The monitoring system is un-calibrated and, by default, the system will fold a large string of transmission sequences into the sampling matrix (Figure 3.7b). All matrix coefficients have ten degrees of movement in this matrix. The largest possible range of a two by two determinant matrix calculation value is ±91 (10x10-9x1); this marks the end of a transmission sequence. In this sampling matrix, there are twenty possible sliding positions between the original and repeat packet transmission sequence (which is either on top or below the original transmission sequence). The difference between two packet instances in this transmission sequence is one. The second highest determinant calculation is ±8 (9x2-1x10), this indicates the number of positions the sequence is mismatched by. The matrix is shifted by one matrix coefficient until the lowest determinant value is achieved (zeroes) (Figure 3.7c). Positive eight shows the transmission sequence is delayed by eight time slots,

and negative eight shows the transmission sequence is early by eight time slots. When this sequence is repeated end-to-end (no pause), resizing this sampling matrix dimension (rows and columns) by this determinant value turns an unmatched sampling matrix (Figure 3.7b) into a matched one (Figure 3.7c). This sequence will become the convolution sequence for the matched filter (a logic number sequencer).

Two new sequences are added into the toy model. One sequence is a packet sequence inversion (10, 9, 8...) from the original. This inverted sequence is recognised when it is repeated and placed either on top or below the inverted sequence. The other expanding sequence adds ten more degree of movement to the transmission sequence with the same arithmetic sequence progression (one). The original sequence (from one to ten packet increments) is now longer (from one to twenty packet increments). The new sampling matrix uses the longest sequence inside its matrix to match all other smaller variations of this sequence, the determinant calculation still reflects the number of time slots when compared to other mismatched sequences.

In this section, a dataset is used to demonstrate the power of this transmission pattern recognition monitoring system. A real time critical application is encapsulated inside the UDP protocol. This dataset was generated from airfield radar and sent to a control tower. The main difficulty is to select the right perspective. The ideal case is to spot a periodic sender, with a low degree of freedom in packet transmission amongst the high rate of change (*m*) transmission sequence. A single transmission is separated into smaller components by their rate of change. These component patterns are then used to align higher rate of change transmission sequence (high packet instance rate of change) patterns. After later iteration, this process will develop a chain relationship, organising other patterns inside the dataset (matching protocols with IP addresses

and MAC addresses). Higher dimensional patterns (higher level rate of change) are more difficult for alignment because of their large sampling matrices. Figure 3.12 shows the highest sampling layer (variation two) matrix for the overall packet transmission recorded.

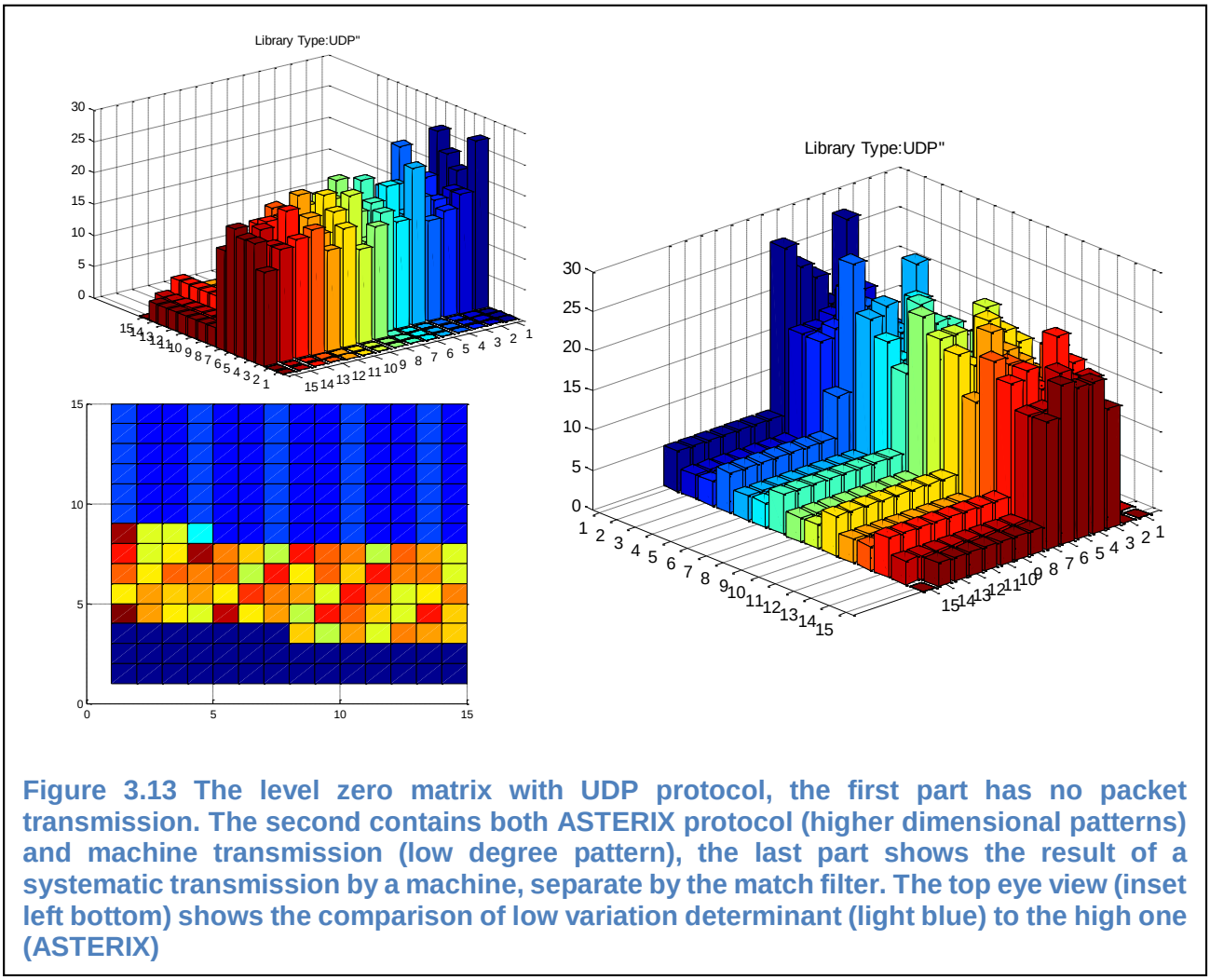


Figure 3.13 The level zero matrix with UDP protocol, the first part has no packet transmission. The second contains both ASTERIX protocol (higher dimensional patterns) and machine transmission (low degree pattern), the last part shows the result of a systematic transmission by a machine, separate by the match filter. The top eye view (inset left bottom) shows the comparison of low variation determinant (light blue) to the high one (ASTERIX)

After having undergone three transformations, the first stage contains blank transmissions in the dataset. The second stage contains UDP packets, which contain both real time critical protocol (ASTERIX) and generic UDP packets for network maintenance. Sampling matrices were constructed by separating UDP router messages from radar, these messages are combined with a rate of change of only two utilising a three by three sampling matrix; a noticeable periodic transmission becomes

apparent when lower rate of change transmissions are combined together. The third part is the reconstruction of all the periodic transmission together using payload per second as the unit (Figure 3.13). Ultimately, this method reveals the ASTERIX protocol which is hidden in UDP.

3.9.1 Discussion

This work has generally assumed that a network is a closed system (transmission variations are gradually stable). Although Ethernet technology allows ad-hoc connectivity ("plug and play"), network transmissions will be stabilised after the initial plug in and setup stage. Ethernet packet patterns provide transparency over senders, routing algorithm and the behaviour of applications over transmission. This method is useful only when the network transmission reaches stability. Transmission sequence calculations are similar to matrix coefficient calculations in a Hankel matrix [142], these so-called *catalecticant* matrix coefficients [143] have properties that can be rearranged orthogonally, and this idea is created to use determination calculations to show the time position and pattern of a periodic transmission, and also by the difference between matrix coefficients. The network category relationship leads to the discovery of Markov chains [144] within devices, network addresses and protocols, where periodic low rate of change transmissions can be compressed into higher rate of change deterministic arithmetic progression sequences. Higher rate of change sequences such as geometric, homogeneous polynomial sequences are less transparent in this design. This research links to statistical and adaptive learning, where support vector machines (SVMs) can be used for sampling matrix alignment,

and statistical Bayes conditions for rate of change approximation [145] . Vectors are optimised to further improve the accuracy of aligning the sampling matrix with transmission sequences. The number of packets (rate of change), statistical distribution and weights are the confidence intervals between deterministic patterns and approximate patterns, even when the matrix size is less agreeable than the rate of change (Figure 3.14).

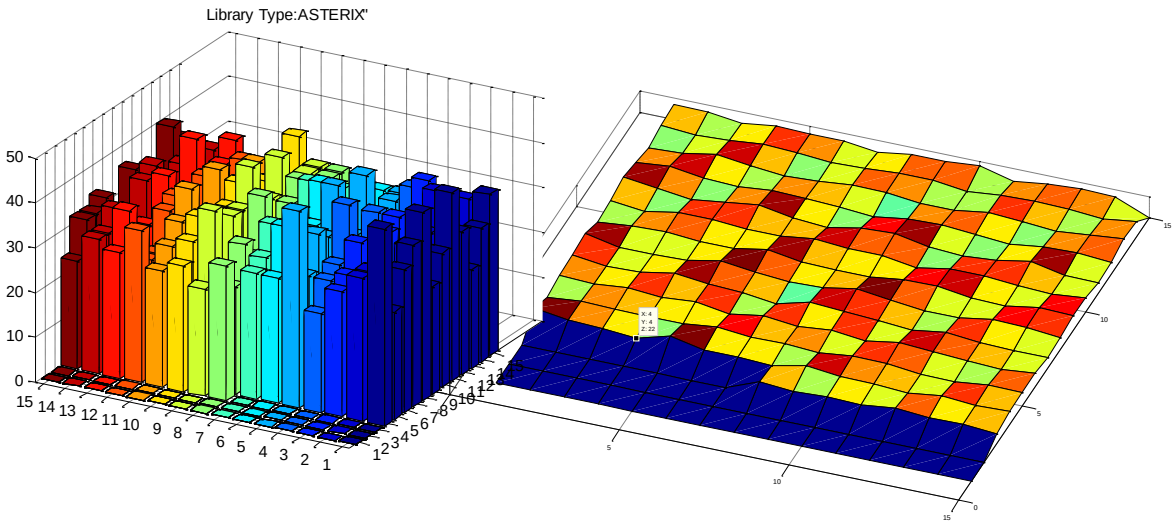


Figure 3.14 The final result is an isolation level two matrix using the pattern previously observed, The diagonal line shows the start and the stop of a sequence (gradient)in the perspective of the machine transmission (low degree pattern)

3.10 Conclusion

This chapter has consolidated the monitoring of a single peer-to-peer communication link system; often the small congestion delays are tolerated by the radar operator and ignored when there are radar points missing. The correct pattern recognition allows these mini-break-downs of communication (labelled as congestion problems) to be

discovered more readily. This process requires many layers and tiers of analytical work to discover the true application transmission rate (label payload per second). Any changes to payload per second can be used to spot any problems in the network. It has been shown that it is possible to monitor the traffic flows of a discrete statistical distribution characteristic using a novel adaptive design with SVM. Other techniques could be used in the field of data mining (learning sets rather than training sets) and using a pattern boundary rather than a fixed one in this investigation to discrimination further payload characteristics in the time of a dataset. As the traffic increases, a trade-off point between the complexities of a technique is needed for real time implementation. Using the known region models (the line) generated from this research it is possible to identify two payload types and congestion problem by the region lines. Thus, further payload types could be classified as anomalies enabling a significant reduction in the workload of the network engineer to monitor a critical optical network.

Chapter 4 – SWIM Server and Client model

The deterministic Ethernet frame transmission can be achieved by micro-managing networking in switching and routing. (Equation 4-1) shows the number of links required per machine (n) if they were to be connected to each other using bi-directional communication (no networking). This configuration has disappeared for the more economical serial switching technology. (Equation 4-1) is an example of a mesh network design. This is the original approach for machines' communications before the overhead serial system (TDMA) or the expanded OSI network model.

$$number\ of\ link\ required = \frac{\boldsymbol{n(n-1)}}{\boldsymbol{2}}$$

Equation 4-1

This safety critical network becomes impractical when there are now 20 to 30 machines, as 20 machines required 190 links and 30 machines required 435 links, using this configuration. This equation is known as the non-blocking arrangement network, because every machine has a connection to all other machines. Each machine is responsible for their connection link, repair and diagnostic. These connections become difficult to maintain, especially when there are many links in the system, all supported by different manufacturers. This problem is overcome by concentrating all the switching connections into a central point. Each machine has only one link, to communicate with the other machine. The first type of overhead is the switching address (serial address); this overhead is used to keep track of the network connection required. This address is normally an instruction for the switch to setup the relevant type of configuration (cross-point). The maximum channel capacity (the maximum data throughput) is divided by the number of machines connected in half-

duplex. This switched network can maintain half of the connection simultaneously, and only when each connection is mutually disconnected with each other. Network design also factor in running the network economically (the cost of maintaining high connectivity) against performance (accessibility), the blocking scenario is economically favourable but not ideal for high performance applications such as in a safety critical network. Ten percent blocking is optimal networking where each node connects only to the right side of the node, leaving the last node unconnected (Token Ring [35]). However, network management with this overhead is not the only method for deterministic Ethernet. The original concept of network is channel division multiple access.

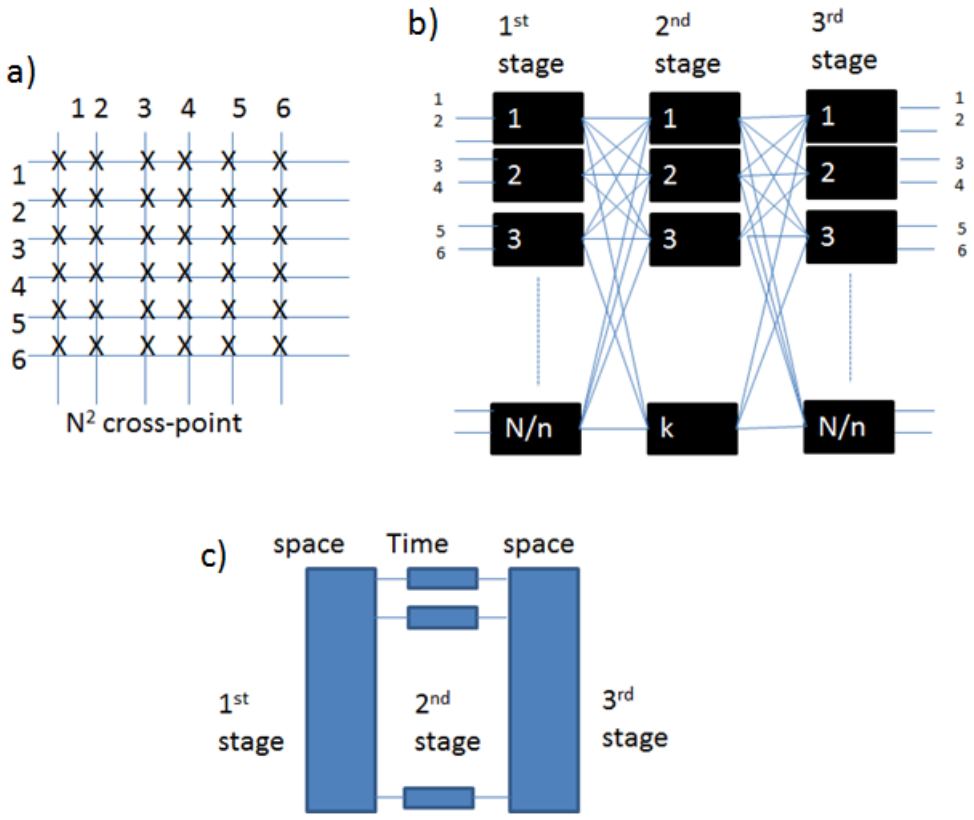


Figure 4.1 Three types of switching configuration possible in a safety critical network (a) Grid cross-point switching arrangement (b) Intermediate cross-point switching (c) Space-Time-Space switching configuration

A channel division circuit-switched network is a management of cross points connections automated without application payload instruction, this requires prior knowledge of node traffic. These connections can also be represented in a square array of cross points (Figure 4.1). A three stage switch uses an intermediate stage (buffer) before connecting each point directly. This is the starting point of having a buffer (holding payloads) in-place of the maximum number of cross-point utilisation. A mesh network has exactly (n) number of cross-points square (n^2). This is easily represented as a square array in which a connection is close by (x) in Figure 4.1. A three stage switch uses an intermediate stage before connecting each point directly. This is the starting point of having a buffer in-place of the maximum number of cross-point. The intermediate stage allows two nodes to be connected and released when the line becomes available. This partition of inlets into groups allows increased efficiency between the usages and number of cross-points. The intermediate point reduces the number of cross points required and increases the number of possible routes to get from one destination to another. Non-blocking versions uses (N) number of nodes/(n) number of connections. Time space switching takes advantage of the fact that each connection is divided up into timeslots [146].

The intermediate stage allows two nodes to be connected and released when the line becomes available. This partition of inlets into groups allows increased efficiency between the usages of cross-points. The intermediate point reduces the number of cross-points required and increases the number of possible routes to get from one destination to another. The three stage switching principle is known as time space switching, which takes advantage that each node is application driven and has low communication utilisation [146]. Predictable switching traffic allows Critical Network Switching to connect links without the need of extra traffic buffers and overheads.

4.1 Routing Protocols

A route is considered to be the best path for delivering messages, the term best is normally refers to the shortest path, with the lowest delay. Often these parameters are considered as cost, the idea is to reduce the cost factor as much as possible. Certain criteria for selecting a route can be the capacity of links, the number of packets queued, the load balance (overloading), security requirement for the links, the type of traffic and the number of intermediate links to cross the node. In an application driven network, routers also use competing network algorithms for dynamic routing. In routing, an IP address is used to select to the payload desirable location. Ultimately, a router maintains a routing table by communicating with other routers and organises the network by iteration (trial and error), which selects the type of service requirements. This network allows single cast (one to one machine), multi cast (one to many selective machines) and broadcast (one to every machine) connection. Route discovery protocols are responsible for creating routing tables, such as Routing Information Protocol (RIP) and Open Shortest Path First (OSPF) [147]. RIP is a distance vector protocol; the cost of a path is determined by distance. This protocol communicates their distance vector table to adjacent routers. This distribution method is called the Bellman-Ford algorithm. Routers exchange information about their link path with other routers. Often the least cost path in a network is around a maximum of two hops (two nodes connection) from each router. This best route database is updated every thirty seconds. In reality, this algorithm operates by hop counting, the worst link (largest cost link) is 15, the hop counter begin with 1, 1 being itself. A cost of 16 represents no connection [41].

Open Shortest Path First on the other hand uses link state; this is often referred to as a link state protocol. The ‘open’ refers to a standard that all the link costs assigned are shared to all the routers rather than neighbour ones in RIP. A router communicates with other routers by ‘flooding’, link state packets are sent in all directions, each connected node in the network reply to the link state packet, each node response determines the link status of their link to their local routers and further router alike. These methods continue until all nodes in the network have responded. In summary, the network topology is known as all routers connected. The link status includes data reliability, data rate, and delays, these are commonly defined in Internet protocol type of service (IP TOS) field. Although there are many other alternative least-cost routing algorithms, OSPF uses Dijkstra’s algorithm [148].

Dijkstra’s Algorithm begins with an initial node source, the best path is selected based on considering the initial step. The end step is to reach all nodes with the lowest total cost. In short these are the steps [149].

Initialise N= source node

For each node m not in N, set $D(m)=c(1,m)$

Find node o not in N

For $D(o)$ is minimum, Add o to N

Update $D(m)$ while nodes still in N: $D(m)=\min[D(m),D(o)+c(m,o)]$

Repeat up until all node in N.

The concept of routing is to create an autonomous network, where routers share routing protocols, and ultimately manage network transmission as a single entity. Autonomous networks use interior routing protocols called interior gateway protocols.

The management of different autonomous networks are managed by exterior routing protocols called exterior gateway protocols. There are two types of hosts; a direct host is a node that directly connects its router to the network, while an indirect host is a node that connects to the network via other means (Switch/hub). The concept of sub-net is introduced here to divide a large physical network into smaller counter parts. An IP address has many layers, a masking area and an actual address to direct messages to their appropriate locations.

A network engineer can design the best route in advance for a small network. This pre-planned routing is called static routing. Dynamic routing uses routing algorithms.

4.1.1 Critical Routing

Network congestion can be removed by dynamically adjusting the available link capacity resources to cater for any unexpected increases in network traffic. This extra network resource allocation system improves the response time for all safety critical application payloads in a network. An unmanaged safety critical network often experiences unexpected network congestion overloads created from network protocol designs such as transport and application connectivity algorithms as has been discovered in chapter three. This chapter contains the simulation of an airport network, where each node experiences network congestion. The current interconnectivity Ethernet adoption protocol makes designing and delivering a real time deterministic network system extremely challenging. The prevailing approach via the Enhanced Interior Gateway Routing Protocol (EIGRP) [41] is presented since it finds widespread commercial application for reliable data exchange. However, it still produces great variation in packet delivery times as a result of the fundamental operation of the

Ethernet system and its philosophy of managing networks on a packet basis. The result of this packet based networking principle gives rise to varying congestion and buffer delay problems. The NTO on the other hand manages network traffic by matching the transmission rates to the link service rate (link bandwidth) in the network. Applications managed in this way create resonant payload requirements as opposed to the arbitrary routing vector metric found using a network discovery protocols. Simulations of a typical local airfield network show that the NTO delivers a reduction of over 25% in network response time, and this may be scaled to much greater in a large network as will demonstrated in chapter 6. Radar data transmissions become not only more punctual but also more protocol efficient through effective network switching without excessive overheads and redundant packets in network switching. The NTO also reduces variations in congestion delay into one deterministic target response time. Moreover, only 1% of packets experience problematic transmission delays compared to 10% using EIGRP – these packets arrive such that their delays exceed the transmission timer and are dropped. The NTO thus offers real time deterministic Ethernet performance for safety critical applications.

This simulation also considers the low delay tolerance factors specified by the EU air traffic SESAR research group server client system, SWIM, which handles all airport information exchanges [19].

The NTO is a concept that combines network switching and routing seamlessly together. These two concepts have fundamentally different operations. The former refers to packet transmission control using overheads - network addresses such as the MAC address for data-link layer, and Internet Protocol (IP) in the networking layer. In reality a transmission scheme only requires an initial addressing operation rather than an address for each individual segregated packet to send data from source to

destination [150]. Current networking processes use external protocols which generate additional network factors concerning the steering of packet routing on a network. Network switching focuses on just the delivery method of network traffic whilst network routing focuses on managing network capacity using external protocols and handling congestion events on a per network basis [7]. Custom applications that use multi-purpose packet protocols such as User Datagram Protocol (UDP) [151], IP, and Ethernet, have no network troubleshooting tools or responses to network faults. The only defense against missing data is left to the transport layer or the application layer, which are ineffective when the missing data are lost as a result of networking issues. A low latency network actual requires an effective networking and switching scheme, and this can only be performed by merging the operation of these two fundamental layers.

A routing protocol is effective at tackling routing problems but it is not designed for managing application transmissions. Transmission requests are left to the transport protocols such as Automatic Request (ARQ), slow start or sample and hold. Network resource allocation problems can be caused by protocol conflicts, this is commonly found between network switching, routing and transport protocols. An example is when a transport protocol such TCP chooses congestion control that is incompatible with the routing decision protocol; this often leads to higher congestion rates with very poor response times [151]. The issue of excessive network congestion arises because these two protocols draw from different stimuli. Routing management protocols such as Internet Group Management Protocol (IGMP) are based on network vector metric calculations [152], while TCP is based on acknowledgement feedback control between source and destination. The impact of these problems escalates for safety critical transmission requirements such as those prevailing in the SWIM network architecture

[19]. This chapter presents an investigation of this issue and aims to find a means to considerably improve the response time of a safety critical application. The network is further improved by effective traffic management in network routing and switching. Section II briefly describes the Enhanced Interior Gate Routing Protocol (EIGRP) [152] which employs vector metric calculations to shape both routing characteristics and network traffic characteristics.

4.2 Server Client Model

In SWIM [19], the server client model is the basis of managing a connection. This model has one server and one client. The connection management model often uses a “use-case scenario”. The client transmits an information request packet to the server, which then processes this request and transmits the information back to the client, depending on client access level. The current Ethernet model has unknown arrival and serving rate to allow flexible support of different services. This unknown service and arrival rate is caused by two types of delay (Congestion and buffering delay) for applications and their unknown payload requirement. This unknown factor has caused many switching and routing issues in packet switching, and created an additional transport layer in the communication OSI model [7]. The actual relationship between arrival and service is the management between packet per second and payload per packet. As far as the switch/router is concerned, it organises traffic by observing the available resources of the link capacity in order to delegate all the appropriate resources to cater for any real time critical transmissions. The perspective of splitting each application into its individual server client model helps to clarify the required level

of packet switching needed to fulfil their designated quality of service, or response time expectations. In essence, critical networking promotes a zero latency network (no additional latency is created in the presence of network congestion). Zero latency networking is achieved not by using packet overheads like the current model suggests (decreasing buffering delay), but by carefully observing and managing the bandwidth requirement of each application and creating a workable traffic shaper scheme that is within the maximum link capacity. To demonstrate this principle an example node with three components: an arrival rate, a buffer and a server is considered as can be commonly found inside a network multiplexer (Figure 4.2).

$$Packet\ in\ queue = (\lambda - m)t$$

Equation 4-2

When there are 5 packets per second (λ) arriving in the node with a fixed 100 bit of payload per packet. The link capacity requires supporting 500 bit/s (m) of bandwidth. The level of expected delay of a packet can be calculated if the arrival rate and link capacity are transparent. For example, when there are 5 packets arriving at the same time, but only 300 bit/s of bandwidth is available in the link capacity (service rate of 3 packets) the extra two packets will be stored in a buffer (assuming the buffer is very large and lossless). This congestion delay will grow linearly at two packets per second with the initial condition (buffering delay) in conjunction to the time instance (t) (Equation 4-2).

In Figure 4.2, this particular form of model is known as deterministic arrival rate in Kendall notation. The first letter “D” is short for “Deterministic” or “degenerate distribution”, degenerate distribution is often viewed as constant arrival rate. Critical

Networking promotes a deterministic variable arrival rate (a step sinusoid wave) and should also be classified as “Deterministic”, the opposite being “M” for “Markov” or “Memory-less”. The second letter is the characteristic letter of the service rate, which here can be D or M. Memory-less arrival and service rate are averages and only packet delay in terms of unitisation can be worked out The third is the number of servers and the last is the queue size [153]. The later sections here explore the difference between multiple deterministic arrival rates with multiple servers. In this discrete example, the number of servers is considered as an increase is service rate.

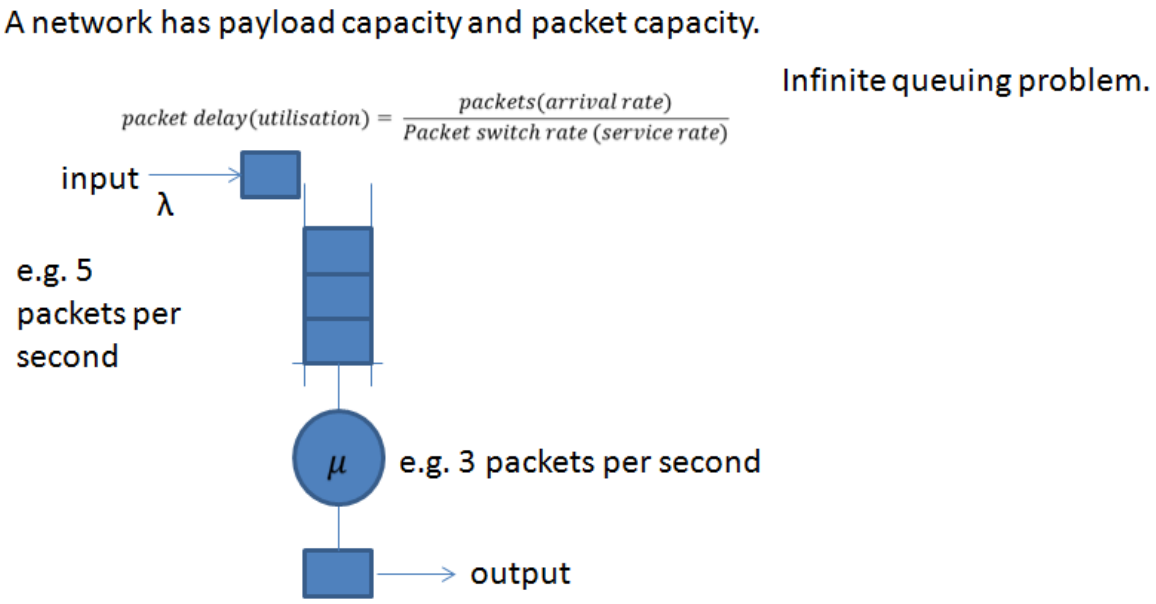


Figure 4.2 Server Client Model to determine packet delay rate based on time instance

4.3 Packet Multi-Path and Truncation

An air field network uses a mesh network, chosen because of its multi-link redundancy, when one link fails, a redundant link can reconnect simultaneously without any hesitation. A mesh network has nodes where multi-point communication paths (multiple arrival rates) combine into one path, this is known as packet trunking. A mesh network also has nodes where there is one arrival rate path, but multiple exit points (multiple service rates), this is commonly known as packet multipathing. In an unmanaged deterministic arrival and service rate network, the delay of packets is the sum of all incoming arrival rates against the sum of all service rate (Figure 4.3).

$$packet\ delay(utilisation) = \frac{packets\ (arrival\ rate)(\lambda 1 + \lambda 2 + \cdots + \lambda n)}{Packet\ switch\ rate\ (service\ rate)(m1 + m2 + \cdots + mn)}$$

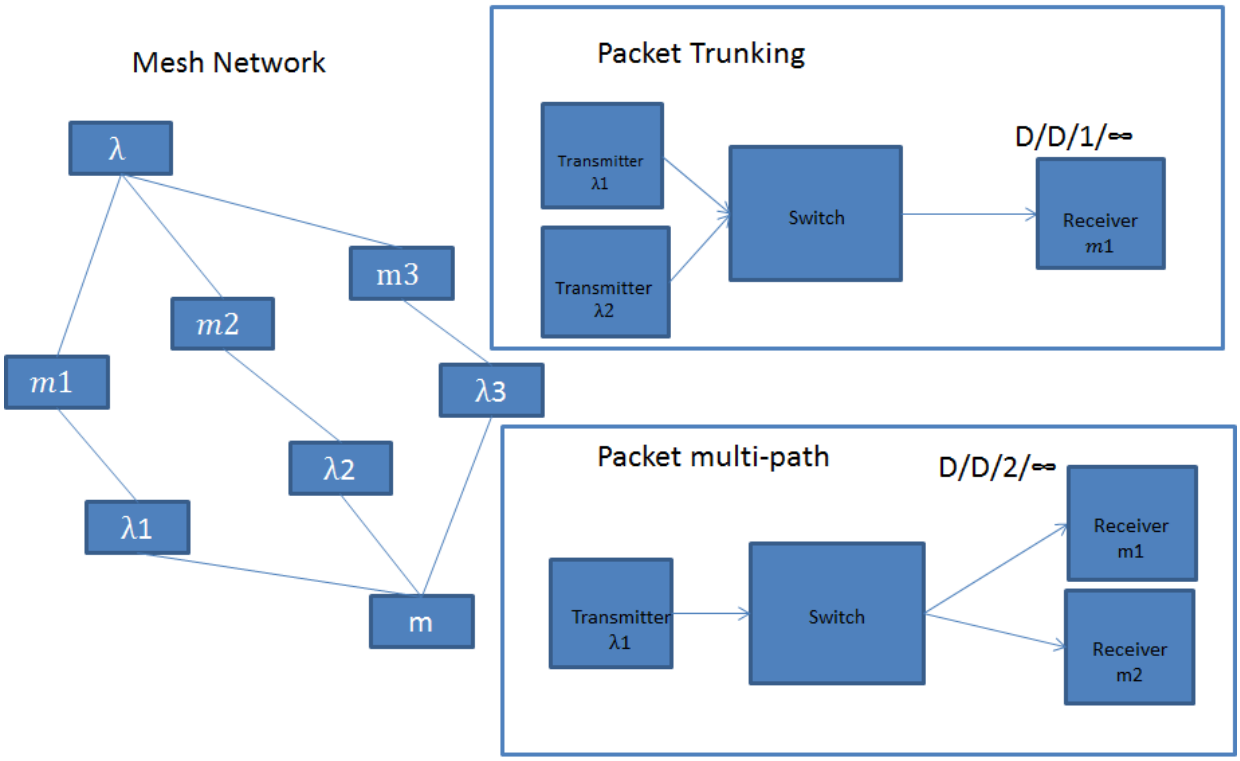


Figure 4.3 The different between Packet trunking and multipath network

Multipathing is a condition where there are an excess of service rate in a node. When multipath routing is managed correctly, a node can transmit over other paths to improve its connectivity. Trunking is a condition where there are multiple arrival paths going into the node, and often is the cause of extra congestion delay from fixed service rates when there are sudden unexpected increases of workload in a network. A node can also have a combination of multipathing and trunking problems. Network mismanagement between multipathing and trunking co-exists in an Ethernet network. This can be further studied using the principle of critical networking parameters which clarified the cause and effect of congestion and buffering delay. Each delay incident can be modelled using the node model as long as the multitude of service and arrival rates is known. The calculation in (Equation 4-2) monitors the net delay either for a group of packets in batch or individual packets depending on the exact time incident and the state of the node. As long as the time frame of each packet is known, their delay can be workout in relation to other packets. Eurocontrol [86] has proposed the used of other networking protocols such as multi-cast, broadcast which improve connectivity using an Erlang Trunking perspective (using a cumulative packet transmission approach of sending more packets for a higher probability of packet reliability) but actually this can disrupt the balance of network organisation by bombarding the network model with redundant packets. Multipath service rate connectivity improvement is lost due to this network casting protocol method. If the network model were observed using Erlang Network Trunking theory, multi-cast or broadcast appears to increase the probability of arriving at the destination by the designated time by increasing the number of trials, but offers diminished returns when the service rate is significantly below the arrival rate. Erlang Traffic Trunking theory is useful for modelling a network with many unknowns, including unknown service and arrival rates, much

like a telephone call centre where there is no control of call arrival rates and their duration (service rate). In this perspective, the service rate is always exceeding the arrival rate, with a traffic intensity of always less than one. Extra queues (buffers) were added to improve the utilisation of the fixed number of servers (but not necessary the service rate). However, buffers increase queuing delay and increase drop out probability from timeout packets (a packet timer is built into the packet so that there are no past data showing as live data e.g. response time window).

4.4 Traffic Theory

The concept of random arrival and service rate connects to exponential service time distributions [154], commonly known as Erlang Traffic theory, although there are many type of random arrival and service rate distribution. This section looks at infinitely variable arrival times (continuous distribution) but with a fixed number and type of packets arriving (discrete distribution). When the network has also a high bandwidth link with a large number of application packets, this can create high delay variance (between the mismatch of service and arrival). This Traffic theory has been revisited because of the similarity of discrete time arrival distribution traffic characteristics (Poisson) from delay congestion in airport traffic data from chapter 3. Individual statistical distributions can be redrawn to look at each causality effect of arrival rate change based on packets received. This work is based on the traffic arrival and service pattern recognition in chapter three which led to the invention of the NTO. Packet blocking probability is the highest fault condition in safety critical communications, and this section will primarily observe this issue using an exponential service time traffic

study in three control dimensions (time, packet arrival and packet service). Each individual change in arrival and service rate increases or decreases the probability of congestion occurrence (which fundamentally increases congestion delay). Blocked packets create missing data for real time communication and could potentially be life-threatening. Blocking occurs from buffer packet overflow and an over congested network (low service rate). Blocking is the break-down of communication and should be fixed at the bottom switching and network layer [155]. The current networking paradigm relies solely on the transport and application layers for retransmission and recovery; the current SWIM method introduces more delay (retransmission delay) from other retransmission protocols. Worst still, delayed packets are useless for the receiver when packets have exceeded their application response time window. So it is essential to transmit data punctually and accurately to avoid packet blocking and dropout. Network planning can be improved by controlling transmission in these several stages: when the type of information exchanged is known, when the traffic flow is known, and when the traffic intensity is known. The blocking traffic model has been used in commercial networks because of their low quality of service requirement, and the blocking probabilities are tolerated by application design and users when the network is busy. Accurate networking is made difficult when there are unknown traffic flowing (often the network experience random user based transmission service), which behaves stochastically (bursty) and non-periodically (randomly). Erlang Traffic Modelling is used to design a tolerated network system within the quality of service design of the application. Erlang traffic models each incoming traffic instance as an independent unit. The expected packet unit is discrete (natural number), but their time intervals can be independent between flow rate (arrival rate) and payload size. A probability distribution such as the Poisson distribution is effective at predicting

unknown arrival rate of traffic instances because of their mutually exclusive service arrival rate.

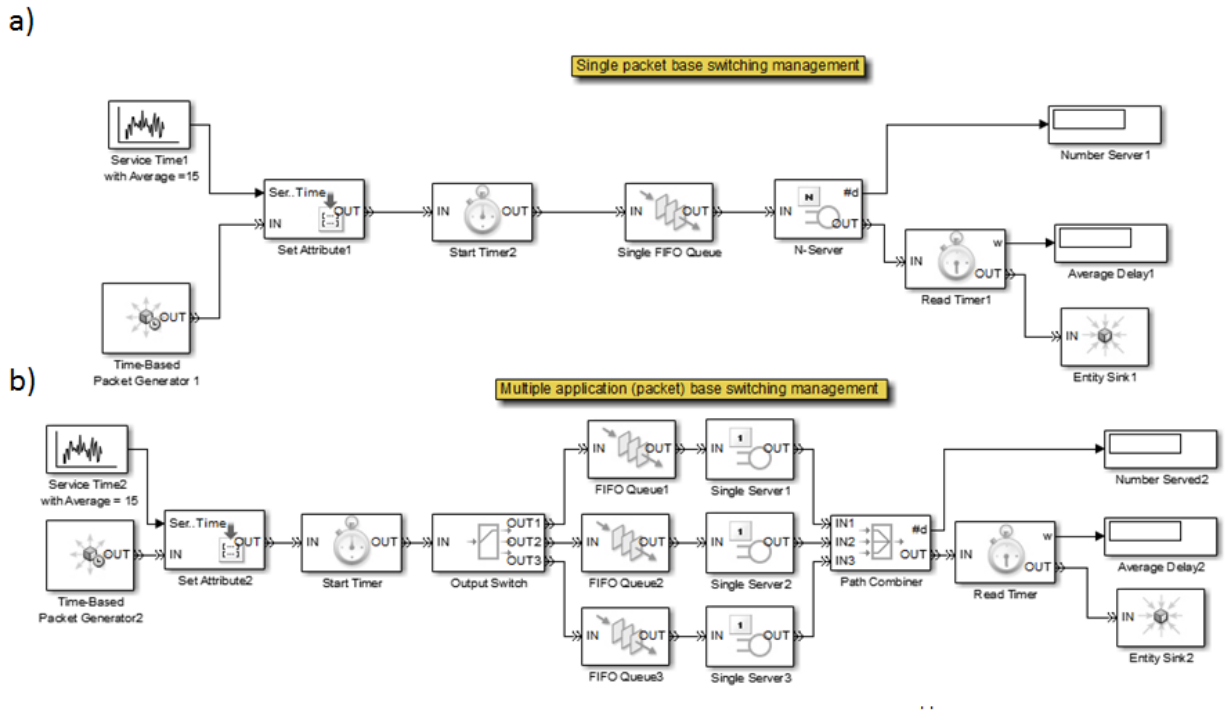


Figure 4.4 The server and queue model of the basic EIGRP packet based routing management. The EIGRP style is labelled as (A) with one queue for all traffic. The multi-queue management system is labelled (B) [152]

These probability estimates can be directly applied to quality of service management to produce a network recovery plan to retransmit missing payload whilst within the available response time window. For example, a safety critical node has (Figure 4.4a,b) unknown arrival and service rate. Multi-queue servers have been abandoned because the conventional Erlang Traffic theory dictates that extra segregated application queues only hinder the operation of higher serving probability and utilisation. This is true, when the network is being observed over the general trend of node usage, however a better service rate (lower delay) can be achieved just by managing the queue actively to the application service demand of each application (a queue for each

application). If each application packet is held for the average of three discrete time samples inside a node, the time between each arrival is five time samples. The arrival rate is a fifth of a time sample. In one hundred time samples, there will be exactly twenty packets placed end-to-end in one hundred time sample, the 21st packet in these one hundred time incidents will be dropped (blocked). Traffic intensity can be expressed as the sum of all the holding times against the packet monitoring duration. Traffic intensity is also refers to the average arrival rate. Estimated average arrival rate can skew the clarity of this study and should be updated regularly. Little's law measures traffic intensity by considering the packet process time (service rate) and arrival rate. If there are forty packets in one hundred time incidents and each average packet processing time is five time incidents, the traffic intensity would be two Erlang. In reality, the arrival times and holding times can also vary randomly too. The Poisson arrival process is closely related to the exponential service time distribution analysed above (between exponential delays), The Poisson packet arrival distribution is the monitored time distribution and the number of packets that arrive [156].

$$P(x) = \frac{(\lambda t)^x e^{-\lambda t}}{x!}$$

Equation 4-3

x is the number of packet arrivals, λ is the arrival rate and *T* is the monitoring period in time incident.

Time critical applications may require a group of packets to function, the above example is the probability density function of individual packets, and could be rendered useless when one of the important application information segments is missing, In an application that requires eight packets to be received successfully within the

monitoring period, the probability of receiving less than eight packets is the sum of all the probability density function trial periods, i.e. the probability of receiving 7 packets, 6 packets, etc. Therefore the cumulative density function of the Poisson arrival rate determines the probability of the number of packets receive if it is higher or lower than the expected packets. Integration by parts is chosen to uncover the cumulative density function; integration takes place with function of a function and factorials. The difficulty of this integration is the factorials, this end up being the incomplete gamma function, with the flooring function of (*x*) within the lower limit of this investigation. The final result ends up as [125]:

$$\frac{\Gamma(\lfloor x+1 \rfloor, \lambda t)}{\lfloor x \rfloor !} \text{ or } e^{-\lambda t} \sum_{i=1}^{\lfloor x \rfloor} \frac{\lambda^i}{i!}$$

Equation 4-4

In discrete simulation, it is only possible to consider each packet and its delay on per time instance basis. For example, an application has an arrival rate of fifteen packets per second at a monitoring interval of thirty seconds, the probability of getting less than eight packets is equal to the probability of getting from no packet to seven packets or 1 – the probability of getting 8, 9, 10, …, ∞ packets. In this theoretical simulation packet are only considered as discrete in appearing at the node and server, but the arrival rate and time period can grow exponentially given that the condition and the distribution result in Equation 4-3 will look different to the discrete simulation. IP fragmentation will have diminished effect, as it expects more packets.

The graph below (Figure 4.5) is the simulation result of a cause and effect study between a packets arrival rate against the number of expected packets. The probability value

is obtained by arithmetic average using one thousand different seeds per data point (30x30). the properties of predicting the probability of having the exact number of arrived packets (x) in the given monitoring period of thirty time sample and the arrival rate of (λ). As the arrival rate increases, the probability of getting packets in the thirty time slots increases, and would be more likely to complete its application packet requirement; a higher arrival rate would require higher service rate to compensate its service. The next scenario analyses batch application packets.

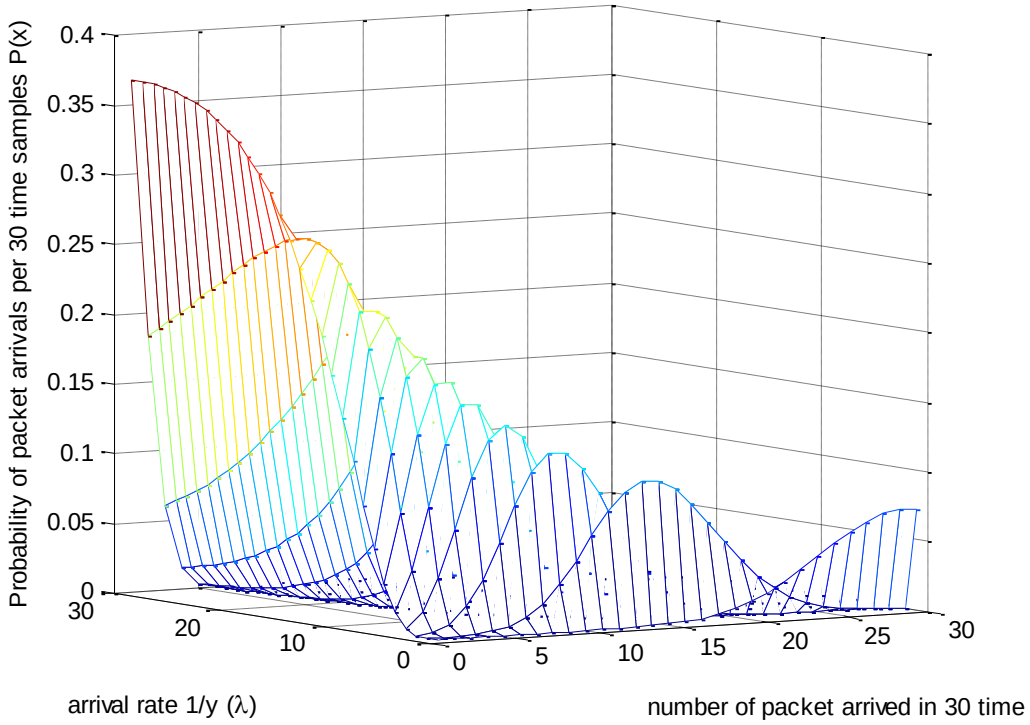


Figure 4.5 The Poisson arrival rate process simulation results in Figure 4.4 The server and queue model of the basic EIGRP packet based routing management. The EIGRP style is labelled as (A) with one queue for all traffic. The multi-queue management system is labelled (B) Figure 4.4

An unmanaged multi-queuing server system would generate the same single distribution as a single system per queue (Figure 4.4b), as the arrival rate and the service rate (Equation 4-5) are random and the two factors can be combined together.

The X-axis is the number of expected packet arrivals which is the from the arriving probability rate. The approximated expression is as follows:

$$P(z) = \prod_{y,x=1}^{y,x} \frac{(\lambda t)^x e^{-\lambda t}}{x!} \Big| y, x = \mathbb{N} \epsilon \mathbb{R}. y, x \ \leq t. \lambda = \frac{1}{y}$$

Equation 4-5

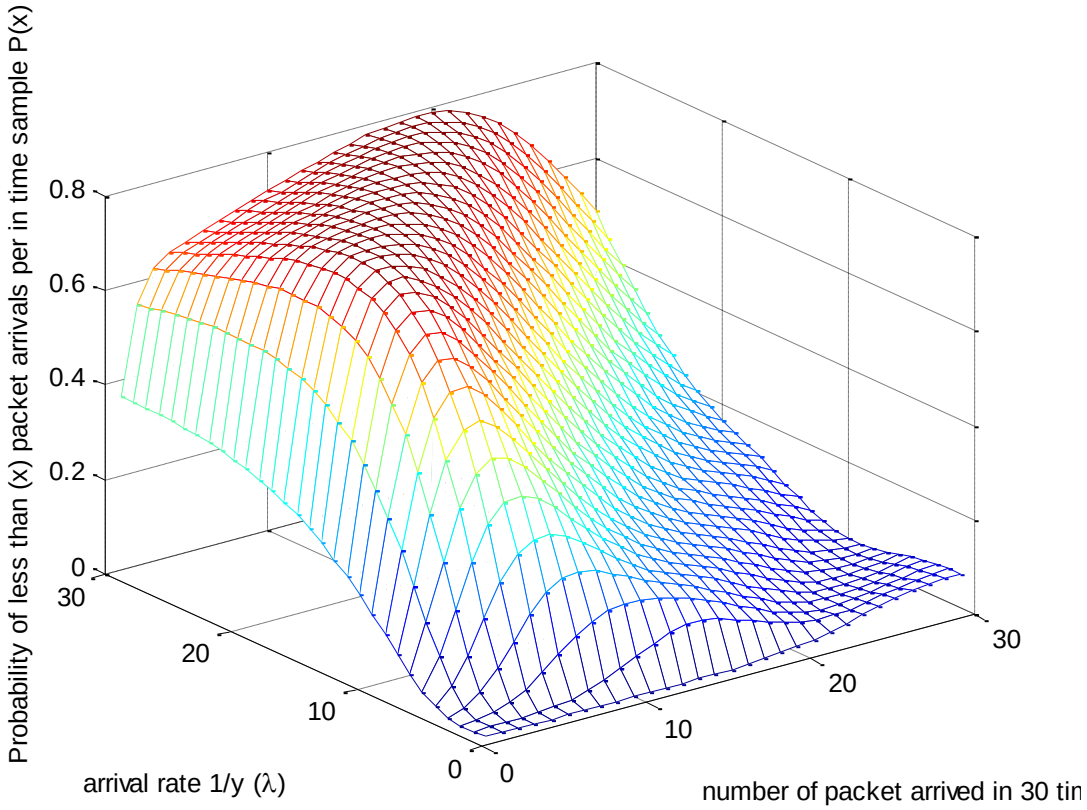


Figure 4.6 The cumulative probability of expected packet in the system given the reduction in arrival rate

A managed multi-queue server is reorganised in Figure 4.6. This graph shows the properties of cumulatively predicting the probability of having the all safety critical application packets from the arrived packets (x) over the monitoring period of thirty

time incidents, the arrival rate of (λ) is given in the Y-axis. The result is fundamentally different from the standard Erlang cumulative function, this time the results of past arrival rates are used as estimates of the next number of expected packets (y+1). Using an intelligent system inspired router (server), the probability of correctly allocating the packet arrival rate are much higher than the standard Erlang cumulative function [154]. The underestimated arrival rate is (y-1), meaning the arrival rate of any value underestimation by the intelligent system inspired serving system. The probability drops exponentially when there are more expected packets arriving while having a low arrival rate (when y is close to one).

The approximated expression is as follows:

$$P(z) = \prod_{y,x=1}^{y,x} e^{-\lambda t} \sum_{i=1}^{\lfloor x \rfloor} \left(\frac{(\lambda t)^x}{i!} + P_{y+1}(z) - P_{y-1}(z) \right) \Big| y, x = \mathbb{N} \epsilon \mathbb{R}. \, y, x \; \leq t. \lambda$$

$$= \frac{1}{y}$$

$$\text{where: } P_{y+1}(z) - P_{y-1}(z) = \frac{(\lambda t)^{X_{y+1} - x_{y-1}}}{i!}$$

Equation 4-6

Similarly, it is important to observe the probability of more packets having arrived than the expected batch of packets (pessimistic view). Originally, this is more difficult, given that the flooring function of (x) and the infinite possibility of more packets arriving from the monitoring onward, however the probability cannot exceed 1 (Figure 4.7). This

indicated that when there are systematic errors in the calculation of the intelligent system inspired router, it can drastically decrease the serving level of each packets.

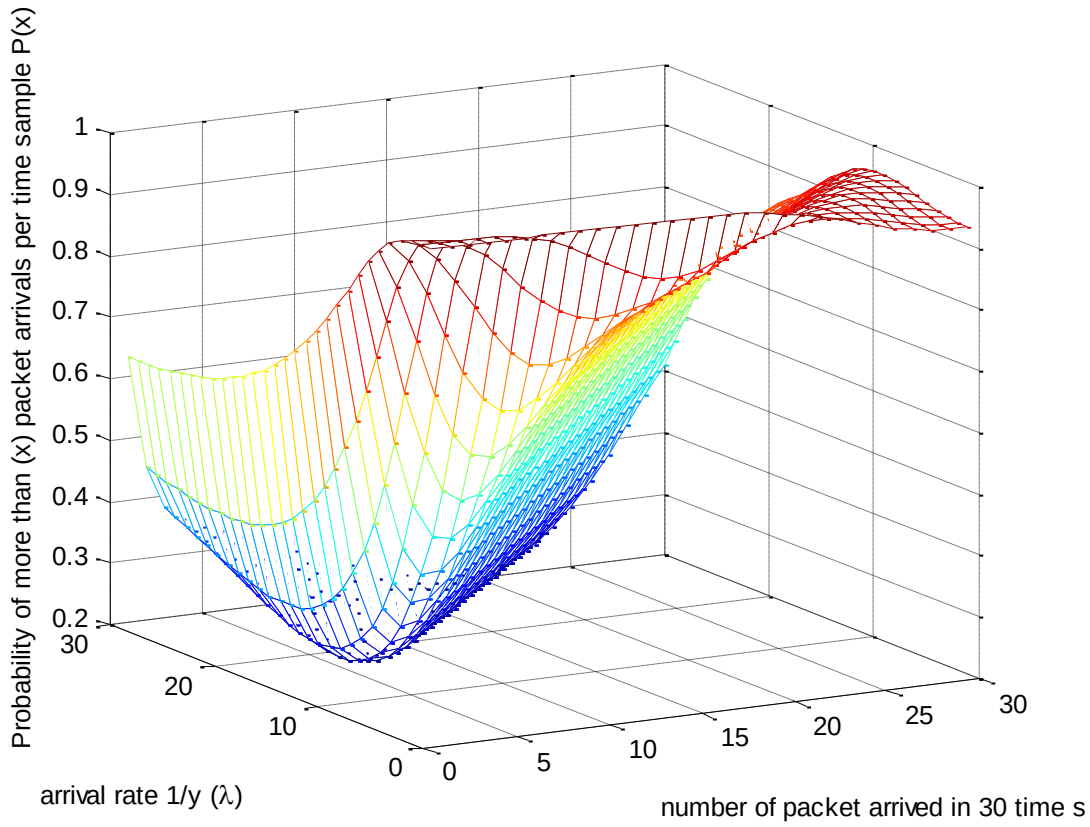


Figure 4.7 The results of expecting to receive more packets (y axis) in the node

All of this is relatively dependent upon the arrival rate of packets and the number of packets arriving in the observed time frame. The Poisson arrival time distribution operation depends upon enough packets not having any relationship with others (independent). Accuracy is assumed by analysis and solely depends upon the average arrival rate (arithmetic mean). The mean has to be both accurate and precise to correctly justify the central limited theorem of the distribution, in-order to budget the grade of service. All of this study is based on the assumption that Nyquist approaches [157] is that the average arrival rate is genuinely at the central limited theorem and the average arrival rate has reached unity in the simulation. This is correct as long as the

study of the arrival rate continuously updates through an extensive period and reaches unity by virtue of all stochastic random events reaching normal distributions. This analysis is more of a descriptive mathematical approach that expresses the overall trends in arrival rate.

The second area of analysis explores the concept of having no packet arrive in the range of intervals. Time intervals with no arrival packet vary continuously and time intervals also follow an exponential random variable in Poisson arrivals time distribution model. Both the mean and the standard deviation are also one over the arrival rate. This can be expressed as:

$$f(\tau) = \lambda e^{-\lambda\tau}$$

Equation 4-7

As this is an exponential distribution, when service time increases (Equation 4-9), the packet arrival rate increases in the node. The completion rate is the time intervals that the node takes to complete processing the packet. The completion rate (service rate) of the packet arriving is measured as the inverse of the holding time of this process:

$$c = \frac{1}{hold\ time}$$

Equation 4-8

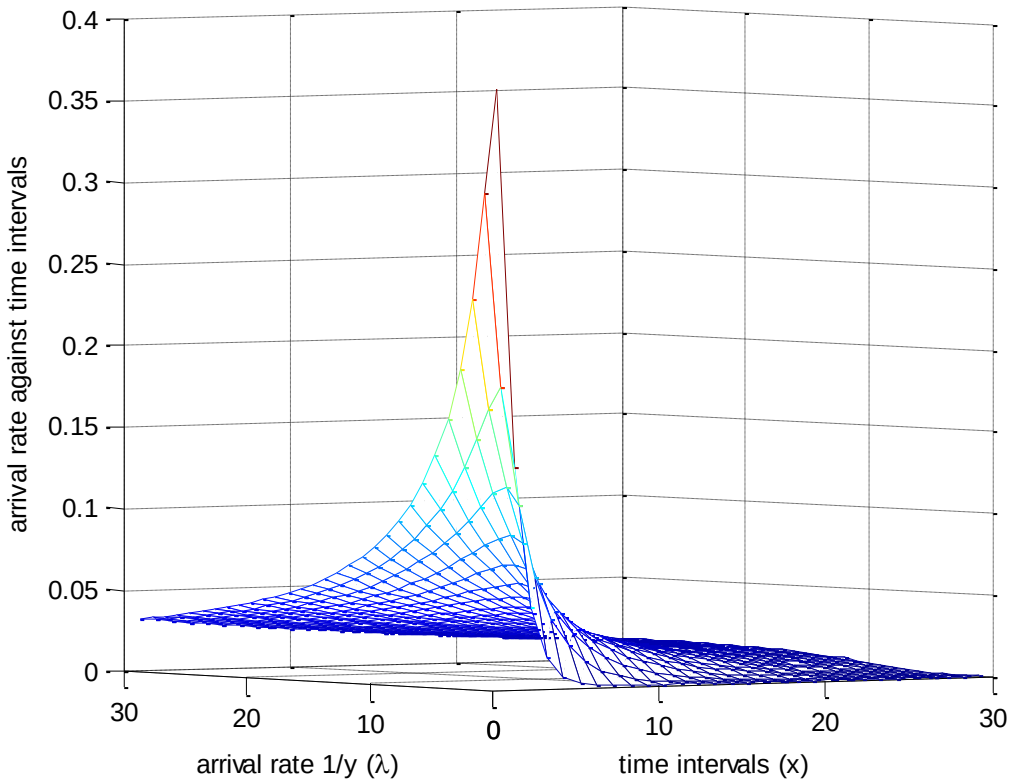


Figure 4.8 The increase of packet arrival rate probability when service rate increases, the time intervals is one over service rate

$$P(x,y) = \prod_{y,x=1}^{y,x} \lambda e^{-\lambda\tau} \Big| y, x = \mathbb{N}\epsilon\mathbb{R}. y, x \ \leq t. \lambda = \frac{1}{y}$$

Equation 4-9

When the service rate exceeds the arrival rate, packets are dropped (blocking). When the packet arrives following a Poisson arrival process rate, a blocking probability can be considered as in three dimensions:

$$Blocking\ CN(b) = \prod_{y,x=1}^{y,x} \frac{\frac{(\lambda t)^x}{x!}}{\sum_{i=0}^{N} \frac{(\lambda t)^x}{i!} + CN_{y-1}(b) - CN_{y+1}(b)} | y, x = \mathbb{N} \epsilon \mathbb{R}.$$

Equation 4-10

The combined arrival and service rate can be considered as $\frac{\lambda}{c}$. When four sources are connected to a node, and this node receives sixty packets arrived per time sample at peak traffic, and each processing time of each packet is two time samples, the probability of blocking experienced by the machine is calculated as approximately 10%. Figure 4.9 considers opening more service points (multipathing) for each extra packet service duration requirement to avoid high blocking probabilities using a traffic model analysed by an intelligent system method.

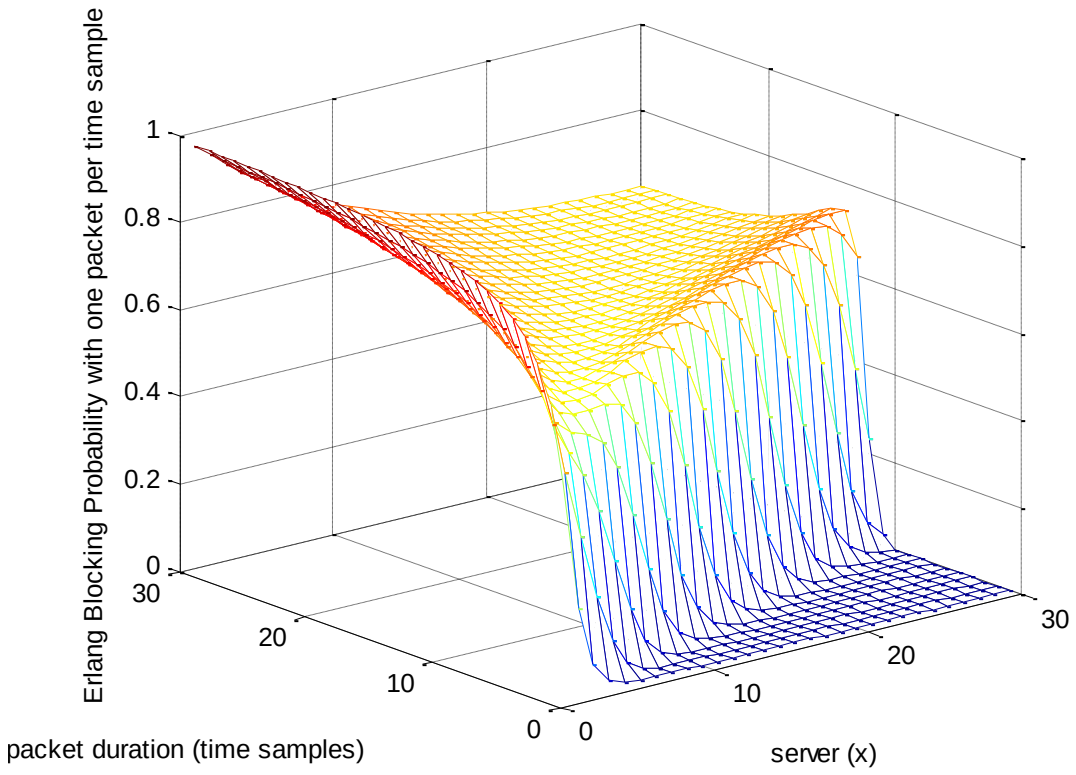


Figure 4.9 The blocking probability of increasing packet service time over the number of servers

The right hand side is the low blocking probability of the server (most packets are served before the next one arrives), the left hand side with the slope is the packet overflow condition. The benefits of opening more buffering points (more queuing slots) are only margin in terms of blocking probability improvements when the packet service rate is higher than the arrival rate. When the combined arrival rate and service duration exceed those of the service rate, a step increase of around 70% blocking is expected with a minor difference between having more server given that the arrival and service rates are unchanged. In conclusion it is more important to manage the service rate than to organised multipathing to improve the overall Quality of Service.

The modern packet based network has many complex problems, created from unknown parameters such as arrival rate and service time. Both parameters are variables with time. A network traffic engineer would normally calculate the traffic conditions over a large period of time and this would result in a normalised arithmetic average over these values. These models are only varied by measuring the grade of service over a long period of time. The general instantaneous relationship between network congestion and quality of service is unsolved. However in this study, the time monitored by the network is in shorter burst, blocking probability is also considered on a per basis of predicable traffic model instead of per random packet basis; this has an improved probability rate for a higher level of Safety integration level with lower break down probability (blocking is considered as a break-down of communication). Queuing, similar to adding extra servers, only reduces congestion delay probability marginally without some direct service rate management. The next section observes the probability distribution of managing service rate, arrival rate and queue sizes with the number of servers.

A realistic network queue can only store finite incoming packets, overflowing packets exceeding the packet queue length are blocked. Networks are rendered useless when packets are consistently blocked by packet overflow. A cyber-attack that exploits this network weakness is called Denial of Service attack (DoS) (Chapter six) producing an over congested traffic network. The blocking probability can be improved by organising the right queue/buffer size. The basic queue can be considered as M/M/1 where both arriving and serving time are Markovian (Poisson process arrival) and there is only one server. This is a classic example of a FIFO (First in, first out) buffer. Packets in the queue may not necessary stay there as each packet has a trip timer. When this runs out, the packet is dropped while in the queue.

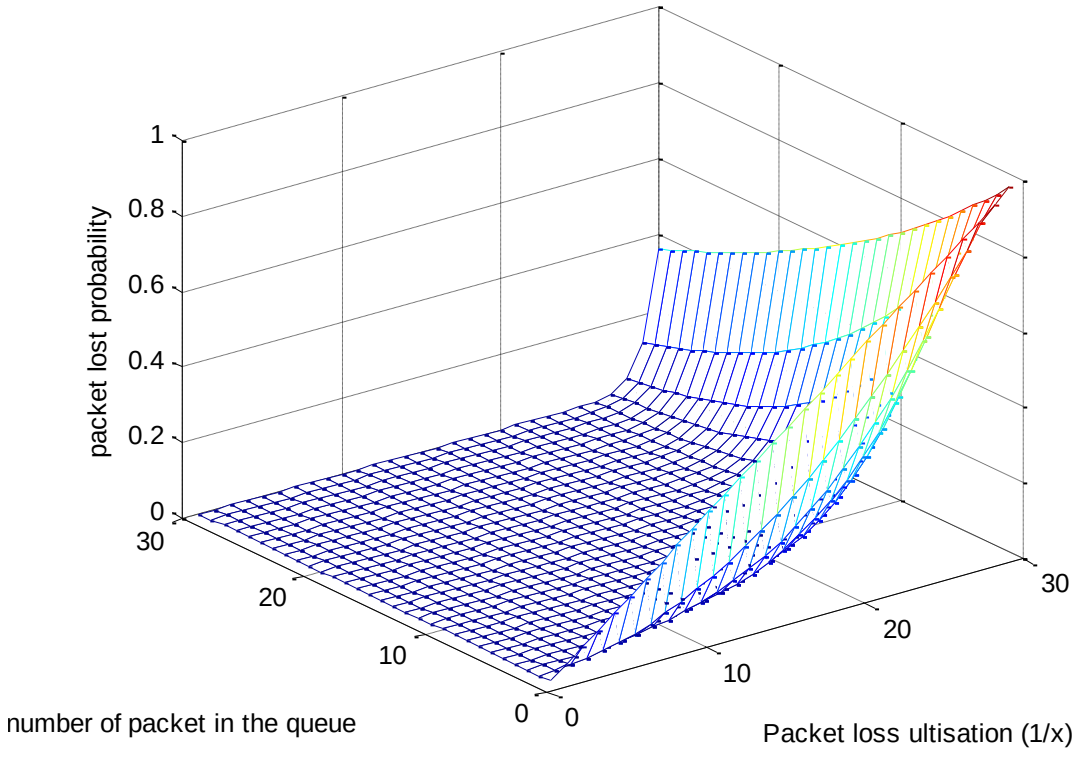


Figure 4.10 The results of dropout probability from exceeding packet trip time in a queue

The ratio of packet loss utilisation is related to the packet trip timer, the higher the lost utilisation, the higher the packet dropouts (Figure 4.10). Finite length queue buffers have an exact value of packet loss probability. (M/M/1/Q) The approximated expression is as follows:

$$message\ lost(p > Q) = \prod_{p.Q=1}^{p,Q} \frac{(1-p)p^{Q}}{1-p^{Q+1}} | p, Q = \mathbb{N} \epsilon \mathbb{R}.$$

Equation 4-11

The Erlang traffic model can be extended to control the queuing in buffer in relationship to the service rate. By allowing more packets to be stored in different buffers and servers, the approximated expressions are as follows:

$$serving\ states\ CN(p > Q)$$

$$= \prod_{y,x=1}^{y,x} \frac{\frac{(\lambda t)^x}{x!}}{\sum_{i=0}^{N} \left(\frac{(\lambda t)^x}{i!} + CNS_{\,y+1}(p > Q) - CNS_{\,y-1}(p > Q) \right) + \frac{(\lambda t)^x}{x!} \frac{1-(\lambda t/x)^{Q+1}}{1-(\lambda t)/x}} | y, x$$

$$= \mathbb{N} \epsilon \mathbb{R}.$$

Equation 4-12

$$waiting\ states\ CNW(p > Q)$$

$$= \prod_{y,x=1}^{y,x} \frac{\frac{(\lambda t)^x}{x!} \left(\frac{(\lambda t)}{x} \right)^{x-X}}{\sum_{i=0}^{N} \left(\frac{(\lambda t)^x}{i!} + CNW_{\,y+1}(p > Q) - CNW_{\,y-1}(p > Q) \right) + \frac{(\lambda t)^x}{x!} \frac{1-\left(\frac{\lambda t}{x}\right)^{Q+1}}{1-\frac{(\lambda t)}{x}}} | y, x$$

$$= \mathbb{N} \epsilon \mathbb{R}.$$

Equation 4-13

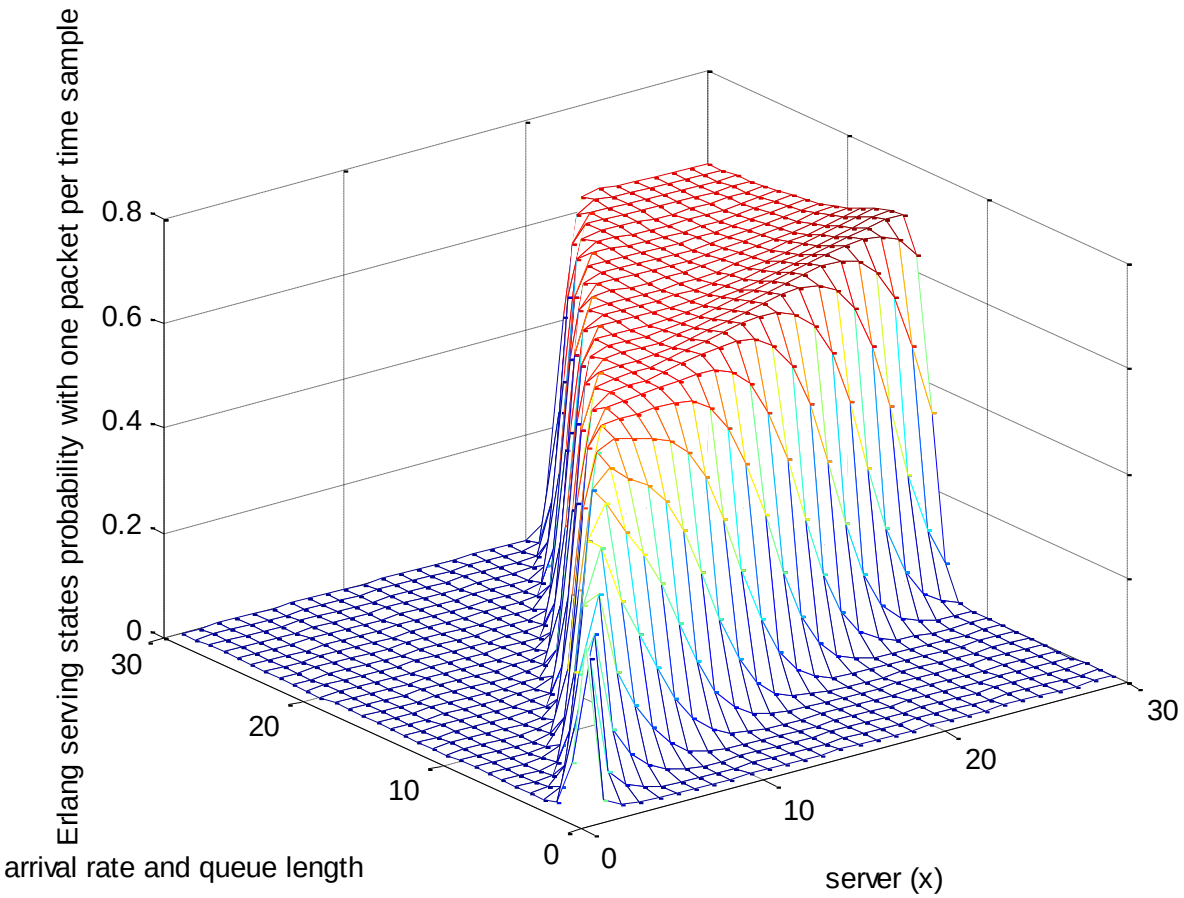


Figure 4.11 The results of serving probability from maintaining queue size and server (arrival rate and service rate matches), to form an area of guaranteed packet serving probability from Equation 4-12

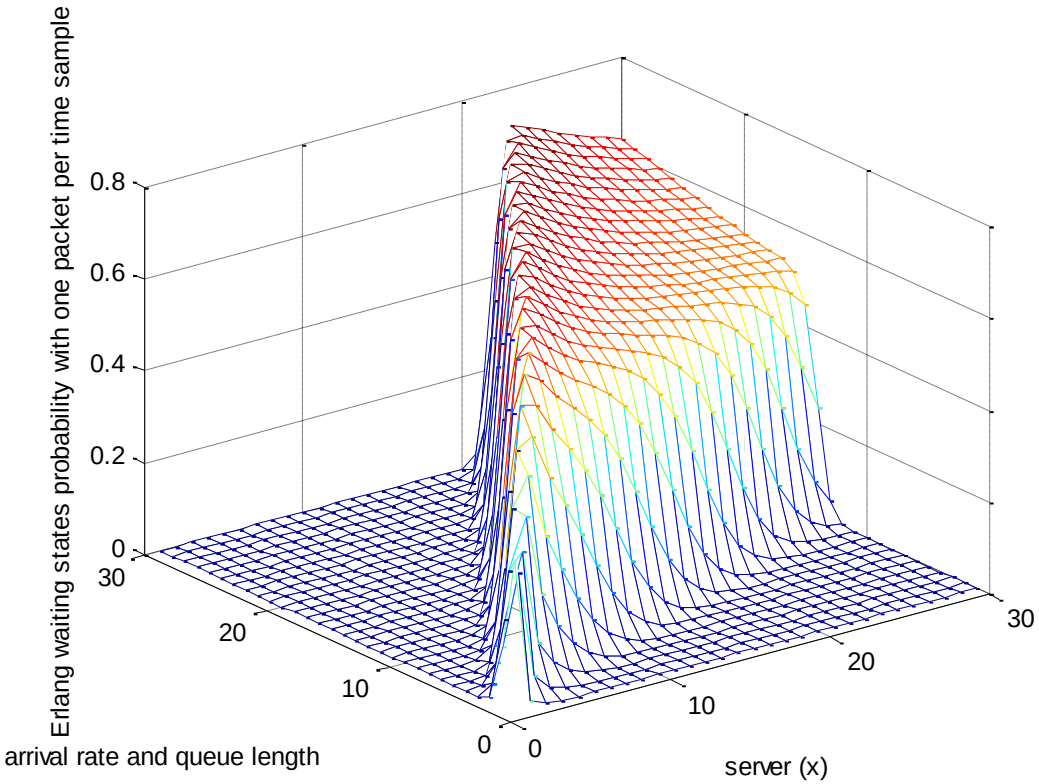


Figure 4.12 The results of waiting/serving probability from exceeding packet trip time in a queue from Equation 4-13, the result here shows a minor improvement when the queue stores packets at exactly the same rate as the service rate (with one more Q+1); a lower excitation level barrier between longer congestion delay can be seen here compared to Figure 4.11

Congestion control is important to maintain connectivity in a network. Packets are managed by flow control (further explored in Chapter five); without flow control, a buffer can overflow, blocked by servers (x) and cause drop out (Figure 4.9). Often the effective communication rate of a network is measured by its bottleneck (the weakest performance link in a network). This protocol prevents the application from suffering prolonged congestion with an individual feedback loop but not on a network basis that is also agreeable to other applications. A protocol feedback loop is a primitive network congestion condition resolver, as it cannot identify the level of congestion accurate by its on demand traffic. The investigation here shows congestion can be caused by many

issues, mismanagement between packet trip timer, queue size, arrival rate and service rate. The largest congestion impact is from the mismanagement between arrival and service rate, and hence the development of the NTO and Critical Network switching. In the past, the service rate of packets has always exceeded the arrival rate, this makes congestion delay small in comparison to the modern delay experienced in safety critical networks, however on a busy network, this congestion problem can escalate out of control. The later EIGRP simulation shows that when congestion delays exceed application response time the problem can escalate to the link being broken.

Since acknowledgement dictates the pace of the transmission process in modern networks, there are two common transmission patterns of slow start and dynamic window size. Congestion control first establishes an available capacity window by sending and receiving acknowledgements, then proceeds by detecting unacknowledged packets with timeouts. In detail, a slow start congestion window reinforces transmission by first advertising its congestion window to the receiver, and gradually increases its rate of transmission by doubling its rate reaching the existing window. When a large number of dropouts occur, transmission is reduced or even dropped until a new updated transmission window is establish. Dynamic windowing on the other hand, just reduces the window size dynamically according to packets and acknowledgements received. In general, slow start is preferable when the transmission application is untimely, but contains critical elements. Any rapid transmission exceeding network capacity causes a broadcast storm. The research here removes congestion control feedback by managing the network resource directly.

4.5 Queuing Probability

The situation is slightly different to the classic one that gives rise to the Erlang C formula where there is a number of servers and a traffic intensity of arrivals. Here, when *n* packets arrive, there are a number of time slots (*x*), created by dividing the available bandwidth by the payload packets.

$$x = \frac{Bandwidth(bps)}{n \times payload(bits)}$$

Equation 4-14

The "intensity" (λ) is the proportion of the slot that is used given the packet rate, number and the payload per packet. Thus the probability of waiting is given by the Erlang C formula:

$$\Pr(\text{Wait}) = \frac{\frac{(\lambda)^x}{x!}\frac{x}{x-\lambda}}{\sum_{i=0}^{x-1}\frac{(\lambda)^i}{i!} + \frac{(\lambda)^x}{x!}\frac{x}{x-\lambda}}$$

Equation 4-15

This formula uses an infinite size queuing system [156] which can create exponential delay when the flow rate (allocated timeslots) cannot satisfy the incoming arrival packet traffic. The NTO is the key for optimising buffer queuing and flow rate size to cater for all incoming packet arrival rates. A large buffer only reduces buffer delay but the traffic delay can be improved with flow-rate management. The workloads

of both buffer and flow controller are monitored by the introduction of payload per packet. There is thus the maintenance of constant payload per packet (fixed workload) in the network, which in turn fixes a uniform delay across all packets. Uniform delay improves connectivity by reducing the need for retransmission from packets exceeding the critical response time. This is especially useful in high traffic bandwidth usage from a high speed network.

Thus the NTO converts the underlying queuing system for packets from one with random arrival rates and random service rates into one where both of these are deterministic. This is because there is a discrete set of flow controller time slots (P_F) with a number of stored buffer packet (P_B) and the NTO fixes the payload per packet rate (R), so that the number of packets in the flow controller and buffer are the same ($P_F = P_B$). Multiple Packets may arrive simultaneously (no intervals between packets), but packet (batch) transmission intervals are fixed with an oscillating payload. This reduces the problem of congestion and buffer delay by maintaining a healthy network workload with fixed response time and provides additional network capacity (flow control) when needed. Application payload varies in all possible routes within the SWIM architecture [19] allowing better routing decisions to be made with each vector metric calculation. This calculation is dynamically updated based on payload per packet workload. In summary, the buffer collects multiple packets to produce a higher payload per packet, and the flow controller will assign more packet transmission slots based on the payload per packet (workload). This is shown in the form of an oscillating application payload with packet. Similarly, the vector metric calculation can be incorporated into the NTO to benefit from multipath transmission. This has the side benefit of dynamically choosing multiple paths for packets for incoming system packets by assessing the minimum achievable delay value of the

metric rather than relying on one path continuously -this naturally reduces the risk of high loss. A critical NTO regulates traffic by adjusting the payload per packet and the number of packets within each available time slot in the physical medium.

Turning to a deterministic Ethernet perspective, this solution only further delays other applications and itself when the network reaches full saturation, i.e. when the network does not have any additional service rate to reduce the delay of the current model.

An Ethernet system is very different however as packets can be modified to have a specified payload size and arrival rate. The only real delay in a network is propagation delay, and there are no real benefits to improving the buffering delay without consuming considerable resources. Critical Networking is proposed to maintain buffering delay as specified by the link capacity unless there is a significant abundance of it. In that case the improvement should be made for all applications, not just certain applications in certain link capacity conditions. Congestion delays are maintained at zero with no requirements to additional transport layer services, so therefore the network and transport layers are close to zero latency.

The secret in Critical Networking is managing payload in the macro perspective and in batches, the rise and fall of the flow rate time incident is easily monitored by phrase shifting of the propagation delay.

4.6 EIGRP Routing

EIGRP uses a vector metric (V_{EIGRP}) to calculate the best route for packet direct transmission. There are a total of six possible parameters deciding the distance of a communication vector metric but only four are employed in the metric [152]:

- The physical link bandwidth (B_E) scaled with respect to 10 Gbps and measured in units of kbps.
- Load - A factor measure with 255 arbitrary levels
- Propagation and response time delay (D_E) in ms.
- A reliability measure with 255 arbitrary levels

$$V_{EIGRP} = \left(\left(K_1 B_E + \frac{K_2 B_E}{256 - Load} + K_3 D_E\right)\frac{K_5}{K_4 + Reliability}\right)256$$

Equation 4-16

This vector metric is adjusted using the free parameters K_1-K_5 to suit custom network routing requirements. Often in commercial networks, congestion delay and link reliability are not considered (i.e. setting $K_2 = 0$, $K_1 = K_3 = 1$ and considering the factor involving K_5 and K_5 to be unity) leading to a metric based only bandwidth and total delay [152]:

$$V_{EIGRP} = (B_E + D_E)256$$

Equation 4-17

Retransmission problems are handled separately by the transport layer. Transport and application problems can be simplified by actively managing routing and switching together to improve the time response of safety critical applications varying K_1 and K_3, and using K_2 for congestion control. Safety critical application packets that fall outside of the transmission window are lost since there is no point congesting the network with packets that exceed the critical time window. The delay factor becomes the important topic in this work, as packet delay in the network layer affects the response time in the application layer. Time critical networking examines the total delay comprising [152]: (a) the propagation delay – the time to traverse the medium (minimum network delay); (b) responsive delay – additional latency introduced by packet retransmission; (c) buffer delay – from the internal packet load of the router buffer; (d) congestion delay – from the number of packets in the system. The last two of these are often probabilistic when the routers are encumbered with random traffic load on their physical links. Here the Erlang C formula is used to encompass both buffer and congestion delay probability in an IP network [125].

4.7 Methodology

The concepts above were tested in a simulation study to design an effective safety critical network for maintaining high quality of service (QoS) across all safety critical information airport transmissions. Here, the particular focus was on the organisation and design of a network linking air traffic radar to the control tower using the SWIM architecture [19]. Information produced from SWIM radar can contain weather reports, air traffic plans and radar information. The initial two simulations examined the traffic

behaviour of the recovery process of three congested physical links, where traffic congestion was severe enough to consider that the link had effectively broken down; the final simulation used the NTO and its vector metric for routing decisions to avoid such problems. The prevailing regime disconnects the physical links when there is a breakdown and this includes payload congestion or transmission time out in addition to true physical disconnection. In the current paradigm, a routing decision is based on a label distribution protocol, and network management. The first two simulations demonstrate destination based and packet based routing algorithms. Destination based routing has a simplified transmission management, but a deceitful label distribution and management protocol could shut down the link prematurely; packet based routing is genuinely a better option for networks with multiple routes but has the potential problem of infinite packet service time due to mismatched or lost sequences of individual packet transmissions. The airport SWIM simulation topology was as shown in Figure 4.13.

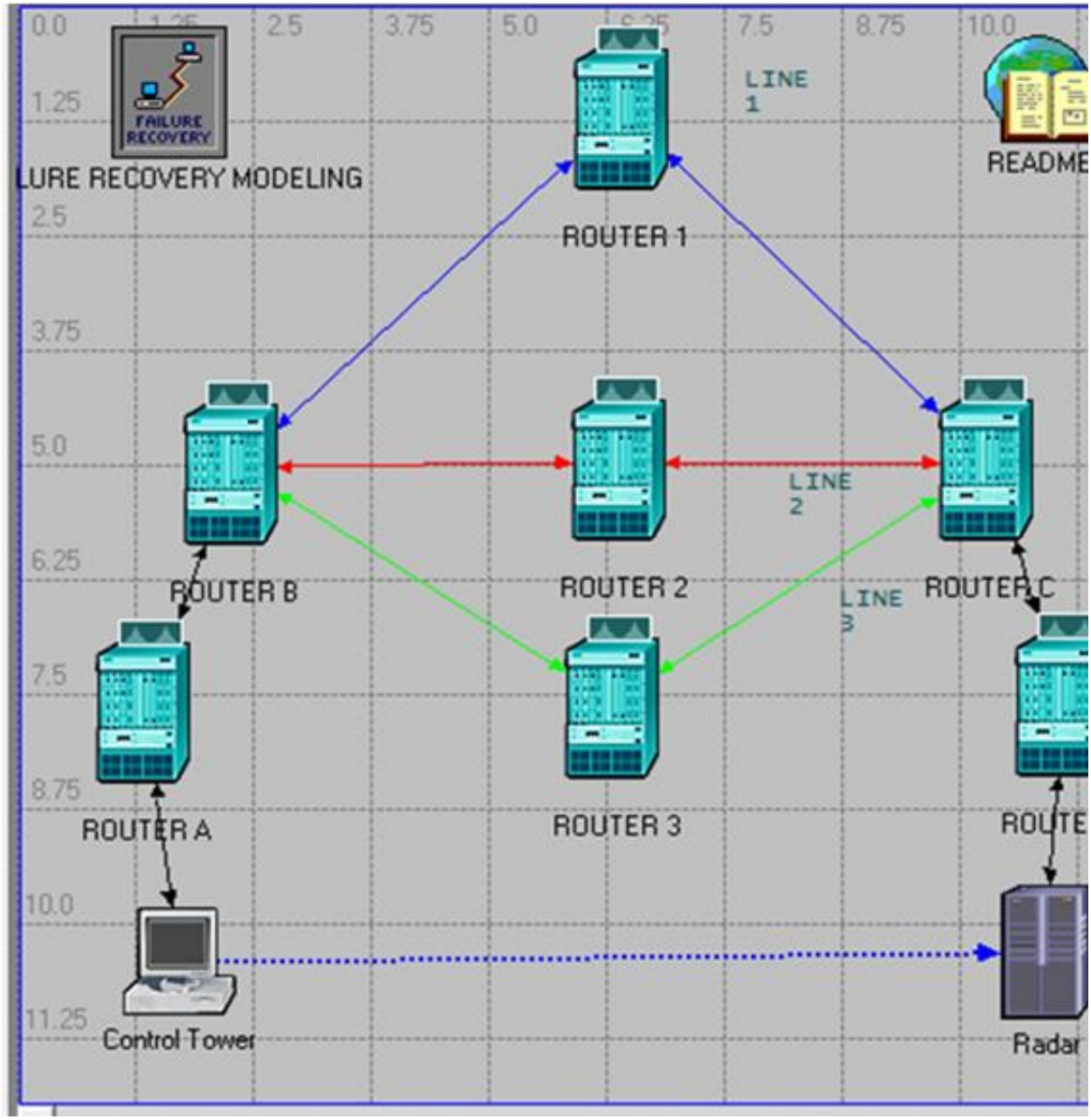


Figure 4.13 Simulation topology in which the control tower is connected to the radar using the multi-path connection via three routers 1-3 in the centre that handle network switching and routing, and routers A- D that handle application information exchange

Multiple routing path links were formed from three central routers to create connection redundancy between the control tower and the radar. Each link had different distances, link one has the longest propagation distance route (blue), link two is the shortest (red) and link three is in the middle (green). All three links operated under 10/100Mbps Ethernet since this represented the typical speed of a radar link and sufficed given the data rate of the radar equipment. The simulation had three stages. The first test was based on the EIGRP routing algorithm with the Constraint-based Routing Label Distribution Protocol (CR-LDP) [158] on a 1 Mbps transmission application with routing service update of one hundred seconds. The single 100% intensity traffic was equally divided between three links in this test with each mid-point router having a first come first served packet queue. Each packet was of size 100 bits and 10000 such packets comprised a sample. The second simulation kept the packet size and number the same but employed a packet based policy for routing with the simplified vector metric (30) to match current EIGRP practice. The total delay had a constant propagation element with variable congestion and buffer components added to conform to the Erlang C formula by substituting an arrival rate based on [156]. The third stage of the simulation implemented an NTO with a vector metric from the air traffic radar to the controller tower.

4.8 Results and Discussion

Here the outcomes of the simulations outlined above are presented. Destination based routing is easy to implement but depends greatly on the operation of

additional protocols. Packet based routing is more dynamic but still faces the issues of random buffers and congestion delay. Only when the NTO manages coherent buffer and flow rate size across the four layers of application, transport, routing and switching can the network achieve precise response times. The NTO also eliminates retransmissions due to lost or timed out packets in the safety critical network system.

4.9 Destination Based Routing EIGRP

Only one link is utilised at any one time and the route decision is set on a network designer basis as may be seen in Figure 4.14(a). Here all the traffic was sent via the blue link, when this fails after thirty minutes all traffic switched to the second choice (green) link and finally to the red link on failure of the green one. Thus, despite the present of three links, only one is used at any one time. This simulation depends greatly on individual link update protocols. Effective traffic transport service can only be maintained from a network with CR-LDP and network management protocols with low protocol update intervals. A dramatic reduction in QoS and reliability results when there is a breakdown between the update time period –here up to 100 Mbit of payload will be lost during a failure. In contrast, a more frequent link service update would improve the QoS in the link by higher link inspection and diagnostics. However, these additional protocols also add more network traffic further increases congestion and buffer delays. This EIGRP configuration offers simpler application transmission management, by using only a single transmission window from each individual routing connection link but at a QoS price.

4.10 Packet Based Routing EIGRP

The second simulation used packet based EIGRP routing, this utilized all three links simultaneously and link breakdown has a smaller effect using destination based routing. The main advantage was that additional traffic flow could be directly assigned to compensate for the missing payload in the system but this design leads to considerable uncertainty in arrival rates. There are instances of packet payloads that are random in both time intervals and in size - this is transparent when the flow controller does not compensate for these changes. Therefore, a strict transmission window management cannot be achieved. The throughput results are shown in Figure 4.14(b).

The maximum application payload is separated and shared across in the three links. Failure of link 1 causes traffic redistribution across the other two links and all traffic traverses link 3 when link 2 subsequently fails. Traffic is re-shared between links 2 and 3 when the former is restored. Each individual packet behaves randomly and this was captured using the Erlang C Formula (18). Each packet acted as an independent incoming caller and the average arrival rate was the maximum flow management rate inside a router. The router flow controller could assign additional capacity (time slots) to cater for the unknown demand load of packets. Unassigned packets were buffered. Depending on the probability of each incoming packet, the total arrival rate of flow management in a router was affected; large quantities of packets with low flow management led to a high blocking probability. This problem would escalate when an application also retransmits timeout packets inside a queue during a mismatch of the application and the flow management window.

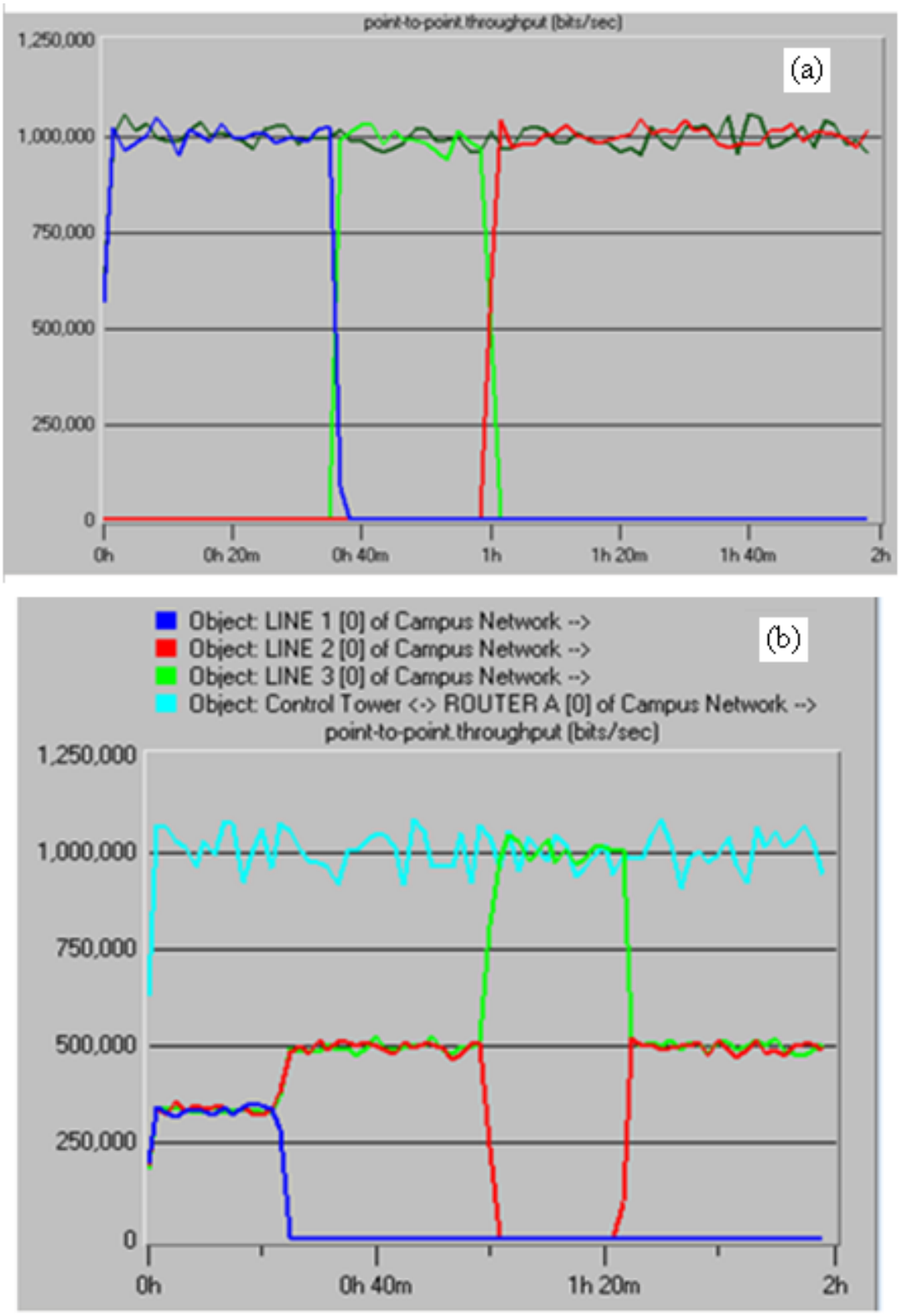


Figure 4.14 link bit rate results for (a) destination based EIGRP; (b) packet based EIGRP

The development of a model based on treating a set of parallel links as a set of parallel servers has been proposed in the literature [156, 159, 160] and so this was done here.Consideration of this design with conventional queuing theory concepts confirmed that a multiple 1Mbit application loads with EIGRP packet based routing closely mirrored the Erlang Traffic C formula. The work load described in this paper used a fixed flow management of one hundred packets (100kbit payload packet), rather than workload bandwidth management as per Mbps on the link.

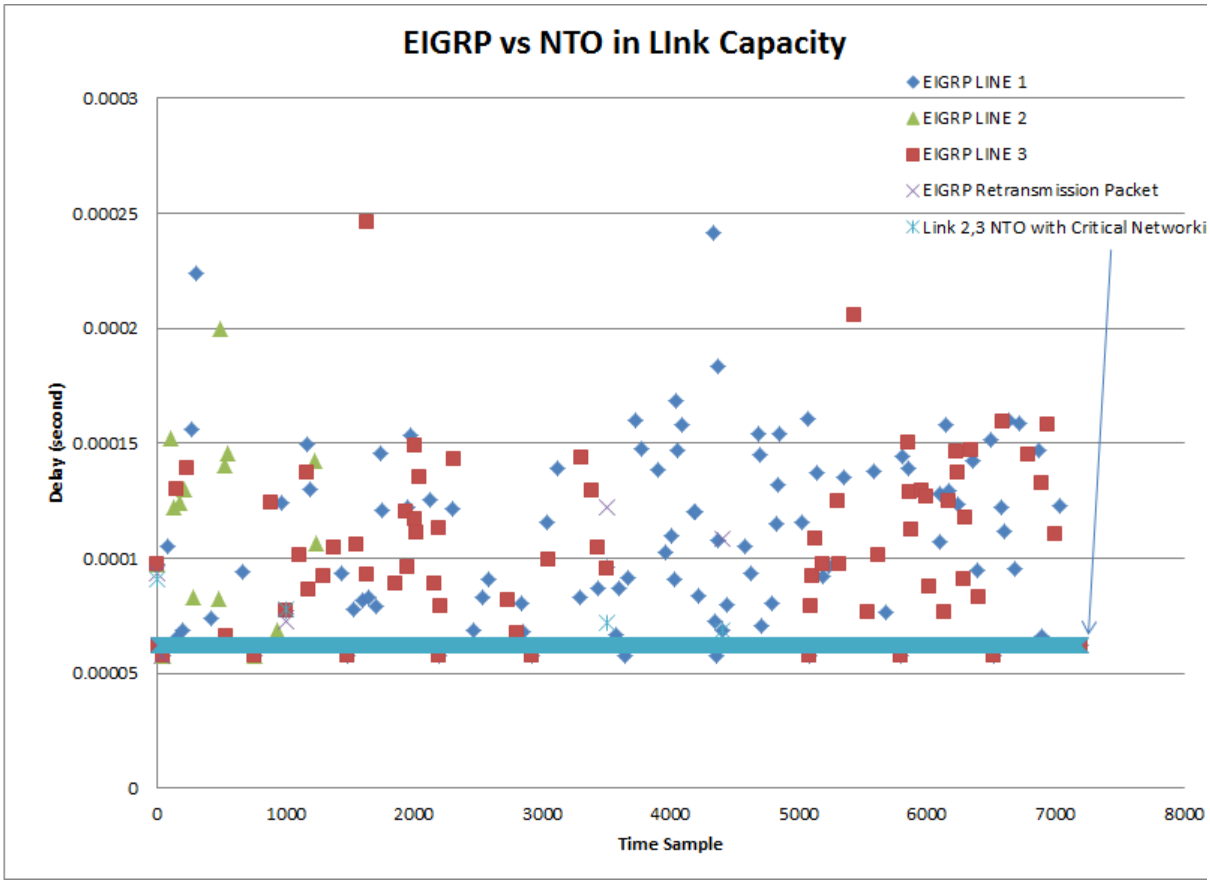


Figure 4.15 Simulation results for application packets affecting the flow management of the routers 1-3. Packet based EIGRP with ad-hoc traffic congestion experiences additional delay of the transmissions and uses all three links. Critical networking uses only routers 2 and 3 with delay affecting only a negligible proportion of the packet transmissions

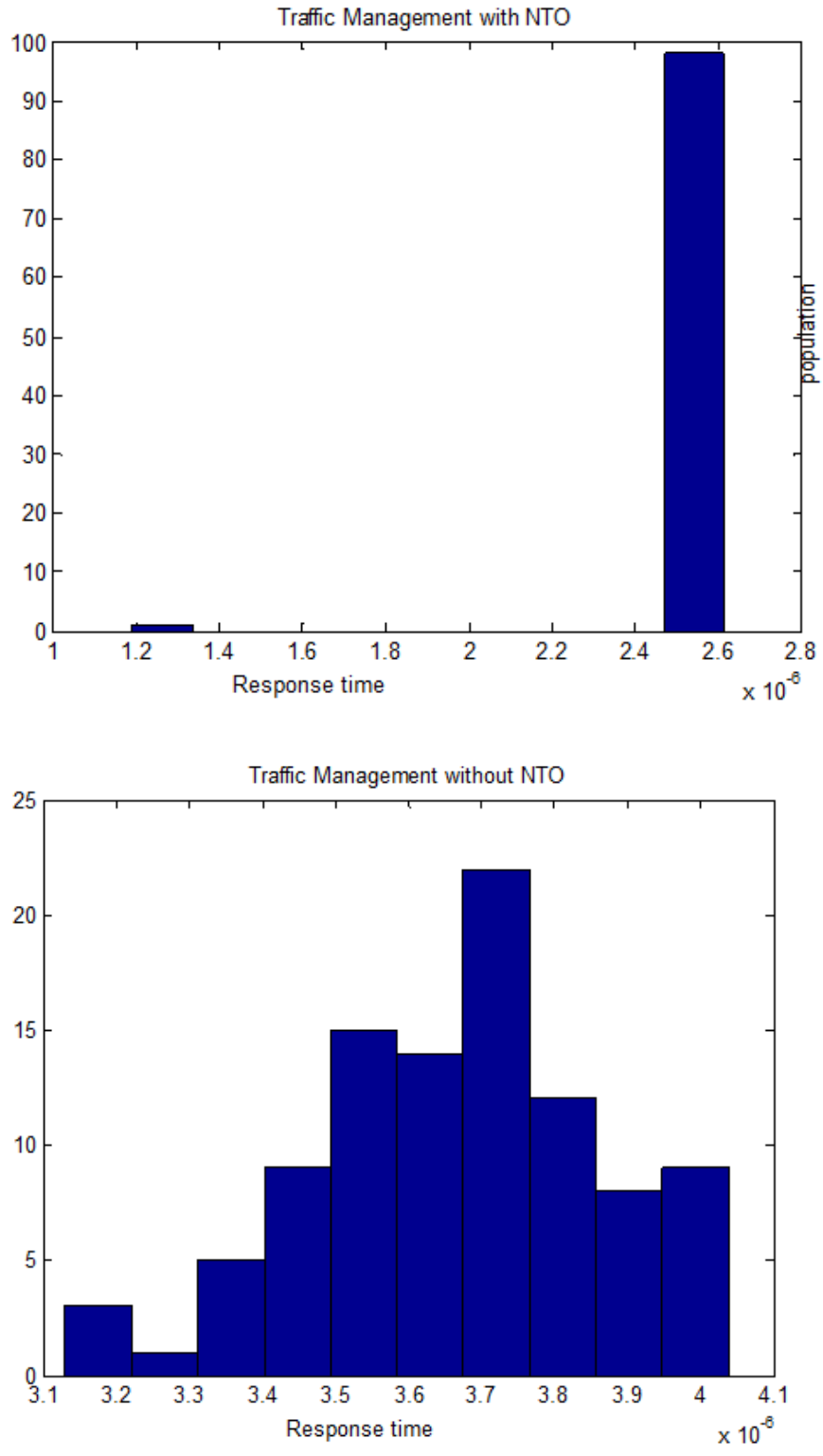


Figure 4.16 Packet response times for: (top) the NTO-based system; (bottom) packet-based EIGRP

The work load was in terms of packets from one to one hundred, this was set as the arrival rate (λ). Each arbitrary flow management slot (time slot) is equally divided from one slot to 100 time slots.

The results for EIGRP in Figure 4.15 demonstrate that when flow management time slots were mismatched to the traffic load, a high level of delay can be expected arising from congestion and delay. The EIGRP lines and retransmitted packets (from instances when the time window is exceeded) shown on the Figure 4.16 clearly illustrate that the delay experienced employing EIGRP is tens of μs more than the results below for the NTO. The number of packets delayed mirrors that expected using the IP Erlang Traffic C formula [156]. The magnitude of the delay time is an exponential decay function, when there are additional time slots implemented during the congestion.

4.11 NTO with Vector Metric

Finally, the concept of the NTO is to divide a single channel link into multiple of maximum packet allocations given at the same time instances as arriving packets in the router. The NTO maintains each application transmission in a strict time management window. A buffer is used for measurement and maintains a constant payload per packet to facilitate the oscillation of payload per packet.The NTO essentially combines all application, transport and routing issues into one oscillating traffic bandwidth. This traffic bandwidth is driven by the application payload per packet parameter. Individual applications are separated by an arbitrary oscillation frequency per IP address and/or per MAC address. Figure 4.14 plots the delays for the NTO

simulation in addition to that for packet based-routing. The critical network matches traffic load with time slots and employs just link 2 and 3 fully.

The multi-variable transmission window problem is reduced and only 1% of packets experience three fixed additional delays. The non-critical network simulation uses all 3 possible links in the setup and delays far more packets. Figure 4.16 illustrates the considerable benefit offered by the NTO in terms of the response time interval between a transmitter and a receiver. For the NTO this time interval is unchanged, multiple packet transmissions are handled in batches with oscillating payload per packet. This is in contrast with the response in the absence of the NTO where not only is the delay some 40% longer, it also has a spread of 25% of its mean value. Thus the NTO improves the radar to control tower QoS to the extent that it may truly be described as *Deterministic Ethernet*.

4.12 Conclusion

Safety critical network investigation focuses on managing an acceptable delay function in application communications. At first, this problem does not appear to be of a huge significance since a small network with a low number of connected devices and low payloads experiences very little delay compared to a busy one with many nodes. However, safety critical networking is currently progressing from small networks to large ones with hundreds of routes and nodes. In this scenario, greater delays are introduced from switching, routing and applications rather than from the physical constraints of the communication medium. In the air traffic context, the introduction of SWIM has led to many modern communication problems, particular in network

switching, routing and application load traffic management. While tools such as EIGRP have addressed these issues with traffic management by configuring an active routing plan (via the K-values and the resultant vector metric) to stave off transmission disasters, these tools are poorly utilised given the modern plethora of application traffic types. The diverse types of application traffic produce a situation that mirrors the communication trunking problem of Erlang traffic theory. The combination of coherent management between packets, payload and traffic is the key theme in this work as a means to reduce congestion and buffer delay during switching, routing and application transmission. The concept of the NTO has been simulated in its application to radar traffic. The study shows how to increase data throughput by considering additional parameters such as the application bandwidth usage, and the synchronisation between receiver and transmitter response time. In comparison with EIGRP, this system offers a reduction of tens of percent in the delay time coupled with a narrow *spread* in this response. Moreover, as a result only 1% of packets experience problematic transmission delays compared to 10% using EIGRP. This is a great improvement in QoS via the improved response time and removal of unnecessary bandwidth wastage from delays and retransmission. In summary, the NTO approach offers not only reliable traffic delivery but also high QoS with minimal delayed or lost packets. It removes the need to add ever greater overheads to network packets and facilitates real time deterministic Ethernet performance.

Chapter 5 – Critical Networking

In communication, application payload control and management are handled by TCP. TCP's congestion control uses acknowledgement (ACK). Without acknowledgement, transmission is reduced or even stopped completely. This protocol prevents the network from suffering congestion, but reduces the effectiveness of maximum transmission capacity by these windows. TCP reduces messages dropped and messages lost by adjusting transmission rate. Since acknowledgement dictates the pace of the transmission process, there are two common transmission patterns of slow start and dynamic window size. Congestion control first establishes an available capacity window by sending and receiving acknowledgement, then proceeds by detected unacknowledged message with timeouts (the waiting to receive time window for acknowledge to arrive in). In detail, a slow start congestion window reinforces transmission by first advertising its congestion window to the receiver, and gradually increases its transmission by doubling its transmission reaching the existing window. When a large number of dropouts occur, transmission is reduced or even dropped until a new update transmission window is established. Dynamic window on the other hand, just reduces the window size dynamically according to messages and acknowledgements received. In general, slow start is preferable when the transmission application is untimely, but content critical. Any rapid transmission exceeding network capacity, causes a broadcast storm. Dynamic window on the other hand, is a parameter that manages the number of transmission through acknowledgements. Unacknowledged messages can either be dropped by the receiver or buffers. These drops occur when messages are overflowed in buffers or receivers. Messages are also dropped when there is bit error [138, 161, 162].

TCP/IP is a connection-oriented service; its connection is maintained by three way communication. The first stage is called a handshake, where both parties agree on a particular window. This handshake is followed by an acknowledge phase and sequence number for each consecutive received packets. Both sequence number and acknowledgement are encapsulated in a transport window. Although this transport method is more reliable compared to UDP, and missing data is retransmitted, it has more overhead and reduces data throughput. In certain scenarios, TCP data could be delayed by up to two window lengths and cause congestion through repetitive acknowledgement and ultimately demanding more of the network resources[138].

5.1 Flow Control

The receiver uses buffers to collect incoming data segments. Flow control is managing transmissions from the receiver; problems include buffer overflow, packet errors and maintaining correct sequence numbers. Sequence numbers are a concept of assigning a number to each data segment being transmitted. A 3-bit binary number can represent eight data segments, label 0 to 7. The modulo-8 configuration allows reuse of binary number (…,6,7,0,1,…) this allows frames to be wrapped around. When acknowledgement frames are maintained in windows of data segment transmission (in blocks), contain more information (frame number error (error control), and transmission rate (flow control)), and send less frequently than the stop-and-wait ARQ, higher data-throughput is maintained. Protocols such as High Data Link Control (HDLC) sends ‘piggybacked’ acknowledgements to both transmitter and receiver. The control field inside a HDLC frame contains, previous information frames, frame error control and flow control [163].

5.1.1 The Three Way Handshake

A reliable connection is setup using three way handshakes; this handshake procedure is a process that sets up an agreement for connection between the two parties (transmitter and receiver). These conditions allow packets to be traceable, whether as lost, delayed or duplicated. In the first step, the two parties exchange their initial sequence number. These numbers are randomly generated by the transmitter. Node A first sends a synchronisation segment to Node B; this sets up the starting sequence number of the communications. Node B then replies by acknowledging the first synchronisation segments and by adding the synchronisation segment with the acknowledgement number. Node A then sends data through this established transmission window by sequence number. When Node A closes the connection, Node A sends a finish bit set with the acknowledgement number to indicate that node A has terminated the connection. Node B acknowledges the receive segments also with the finish bit set and the acknowledgement number, and Node A sends an acknowledgement to finally terminate the connections [164].

Adaptive retransmission is a process that manages errors in transmission and packets dropping from the receiver. This process sets up the procedure for resending erroneous or dropped packets. The reduction of response between acknowledgements reduces the effectiveness of TCP/IP. The two models are immediate response (empty segments are sent from a receiver to acknowledge every successful segment received) or a cumulative acknowledge response (groups of acknowledgement wait before sending).

The two main operations of TCP are positive acknowledgement and window timeouts. Negative acknowledgement is missing in TCP, and may play a bigger part

in the future. A data segment dropout is unacknowledged. Unacknowledged data segments would be retransmitted, similarly acknowledgements that are received after the period of timeout window, would also trigger retransmission. The timeout window period should be much longer than the round-trip time (The time for a bi-directional communicate between the two nodes). A Timeout window period that is smaller than the round-trip time, results in all data segments being retransmitted. Similarly, a timeout window period that is much longer than the round-trip time, results in an inept transport protocol. The ideal timeout window period should be just over the round-trip time, this is both responsible for retransmitting unacknowledged data segments and manageable retransmission. The ideal timeout window is flexible because round-trip time periods are dynamically changing in a networks, the traffic condition in links and router routing algorithm often dictate the pace of the round-trip time. Singularity round-trip time (SRTT) operates ineffectively over IP networks. Early approach to an adaptive retransmission scheme is described in RFC793. The average round-trip time (ARTT) (arithmetic average across many round-trip times) is calculated in this format (Equation 5-1-4), where k is the number of request and α is the average request time [165]

$$ARTT(k+1) = \frac{1}{k+1}\sum_{k=1}^{k+1} RTT(k)$$

Equation 5-1

$$ARTT(k+1) = \frac{k}{k+1}ARTT(k) + \frac{1}{k+1}ARTT(k+1)$$

Equation 5-2

$$SRTT(k+1) = \alpha SRTT(k) + (1-\alpha)RTT(k+1)$$

Equation 5-3

5.1.2 High Level Data Link Control

HLDC is a standard protocol for flow control, protocols based on this move the timeout window based on acknowledgement. The traditional flow control parameters are [166]: Sequence number (SN), Acknowledgement number (AN) and Timeout Window (W). The timeout window period is decided for data segments to be transmitted without the receiver acknowledgement. The issue of overflow is avoided from the receiver advertising a timeout window to the transmitter before sending actual data segments. The window value advertised is also the maximum amount of data segments in a timeout window buffer. The timeout window period remains at standard period if the receiver accepts TCP data segments faster than the arrival rate of the transmission. Each transmission from the transmitter reduces the timeout window period. TCP transmitter waits for transmission when the timeout window reaches zero. The transmitter waits for a new timeout window before the next transmission. The receiver advertises a new timeout window before accepting the last transmission inside the timeout window. Network traffic increases when the transmitter sends out a blank 8 bits data segments during the new advertising window, this is also called a ‘probe segments’, this will most likely be rejected by a receiver, if the receiver responds, this will trigger another non-zero windows.

5.1.3 Automatic Repeat Request (ARQ)

There are two ways to recover from errors, namely correction or sending a request for a repeat message. Errors can be corrected using hard forward error correction (FEC); this is similar to advance parity check in a matrix form and Hamming code. Code for detecting errors uses less code that detects and corrects the codes. In this section,

this study focuses on the effects of retransmitted request to the network system. Assuming the fact that powerful error detection is used (no missed error undetected), the three key types of ARQ scheme are stop-and-wait ARQ, go-back-N ARQ and selective-repeat ARQ [167].

Stop-and-wait ARQ is often used in a half-duplex network. The receiver sends positive acknowledgement if there is no error detected and negative acknowledgement if there are errors detected. This Stop-and-Wait simple network scheme is inefficient because of waiting (two times the full round trip time).

Go-back-N ARQ on the other hand uses full-duplex links. Duplex links give an option to continuously transmit without any pause via downlink, acknowledgments and negative acknowledgments are sent via uplink. Go-back-N ARQ organises transmissions in blocks (a sum of transmission packets), negative acknowledgements trigger block retransmission. Acknowledgements still suffer from a round trip delay. This method is ineffective when there are high error rates, as many retransmission blocks block transmission channels. Selective-repeat ARQ only retransmits packets received with negative acknowledgement. Packets are later rearranged in order by the receiver, this is not suitable for real time order transmission [168].

5.2 Critical Networking

Critical Networking was invented to simplify all complex network issues into one, treating all first packet communications with a guarantee designed arrival time, and operating within the network link capacity limit. The arrival time of a packet is not dictated by the number of packets in the system but decided base on the size of each packet, the *Payload per Packet* (R) and the available link capacity bandwidth (bit per

second) that the network can support. A fatter packet (one with large payload) takes longer to transmit down a link than a thin one; this is known as buffering delay. For example a 10kbit packet would take 10 seconds on a 1kbps link, whilst only 1 second is needed for a 1kbit packet.

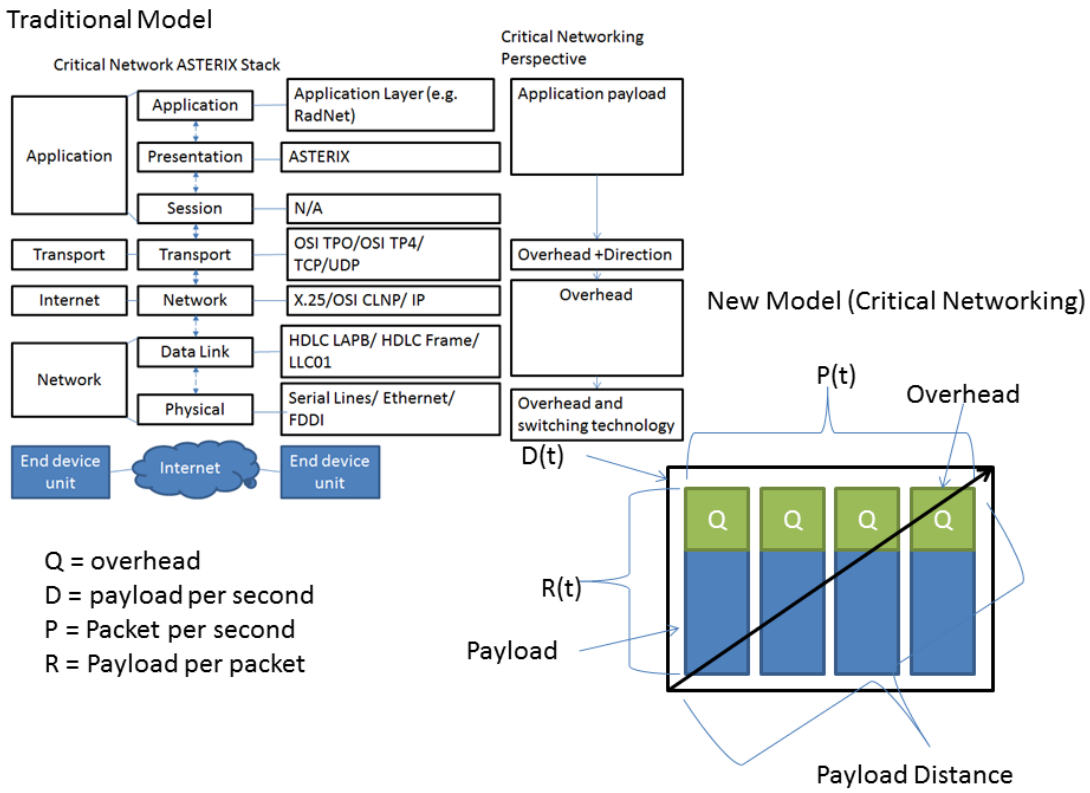


Figure 5.1 This diagram shows the difference between the traditional packet switching OSI Ethernet Packets model and the new Critical Networking Ethernet model

In this model, overhead (Q) is related to the number of packets in *Packets per second* (P) in Figure 5.1(Equation 5-4) below. To illustrate the point and show how the idea of P(t) enters the model, a 2 packet per second source with a packet size of 5Mbit is considered, transmitted using 10BaseT and 100BaseT[169] (i.e. at 10Mbps and 100Mbps respectively). Although it takes 500ms to transfer the payload using 10BaseT and just 50ms via 100BaseT, viewed in terms of packets the actual arrival

rate is two packets per second in both cases. This critical network bandwidth set by the slower of the channels means that although the potential arrival rate is different for the two physical technologies, their actual arrival rates are the same but with different utilization rates. Moreover, when the application transmits over an asynchronous network, a converter adds additional information such as source and destination addresses to each packet. It is thus possible to define a quality factor, Q, as the ratio between overheads and data, for example Q = 2 doubles the payload size and hence the application bandwidth.

$$R_{\mathrm{in}}(t) = Q_1 P_{\mathrm{in}}(t)$$

Equation 5-4

To develop a critical network model the parameters introduced above, namely packets per second and payload per packet denoted by P(t) and R(t) respectively are germane, where packet is used in the conventional sense of information fragments from packet switching [170]. Given their definitions, the product of these parameters is packets per second × payload per packet = payload per second, i.e. D(t). Moreover, P(t) is the familiar measurable quantity but R(t) is the potential payload that is sent as one packet which may be greater that one by buffering small arriving packets. The case when the packet arrival rate is below the maximum transfer rate limitation is the pertinent one here so that the critical networking bandwidth can work independently of the lower layers. So the type of source under consideration here always generates packets at a rate below the maximum physical bit rate (Equation 5-5).

$$\mathrm{D(t)} = \mathrm{R}(t)\mathrm{P}(t)$$

Equation 5-5

Packet overhead only increases each packet payload size and it is not ideal for an application that requires the shortest response time (i.e. time between packet transmissions). Large packet overhead (Q) should be avoided, and only applied when it is necessary. Packets per second (P) is related to TDMA method in data communication, where a packet is treated as a time slot, the arrival rate can be thought of as inverse packets per second. For example, when five packets arrive simultaneously in one second, the packet traffic is five packets per second, and the arrival rate is one fifth of a second.

The maximum packet transfer is when the packet payload size takes up the same time space in link capacity as the arrival rate. A 1kbps payload rate that has 250bit of payload per packet can be simplified to five payloads per packet or five payloads per second. This is the absolute maximum transfer rate, as there is no time gap between packet transmission intervals. When there are many packets payload transmitted within one time incident, the measurement unit for analysing such traffic is payload per second (D). Applications are encouraged using this unit to estimate their designated times of arrival from the transmitter to the receiver across multiple different links in a network. The application link capacity should decide on the way in which application packets are transmitted and should be in unison with the application and network specifications.

Link utilisation is another important concept for Critical Networking; saying that a link has 40% utilisation does not necessary mean either (a) there is exactly 40% of application packets per second in the link (Only true if the link breaks down all individual payload into packet per second) or (b) there is exactly 40% application payload per packet either (True only if the link combines all payload into payload per

packet). The truth is somewhere in between, a 100% packet per second link utilisation could have 10% payload per packet link utilisation - each time slot in the link filled with a small payload packet - thus wasting precious link capacity spaces. Similarly, a 10% packet per second link utilisation could deceitfully send packets on a 100% payload per packet link utilisation, thus packets are blocked from payload buffer overflow. The terminology payload per second link utilisation should be treated as a vector for effective routing and not as a scalar. The routing magnitude of payload per second is called payload distance to avoid any confusion. This is useful for calculating any additional congestion delay experienced in a network by dividing payload requirement by the link capacity.

5.2.1 Overhead Compression

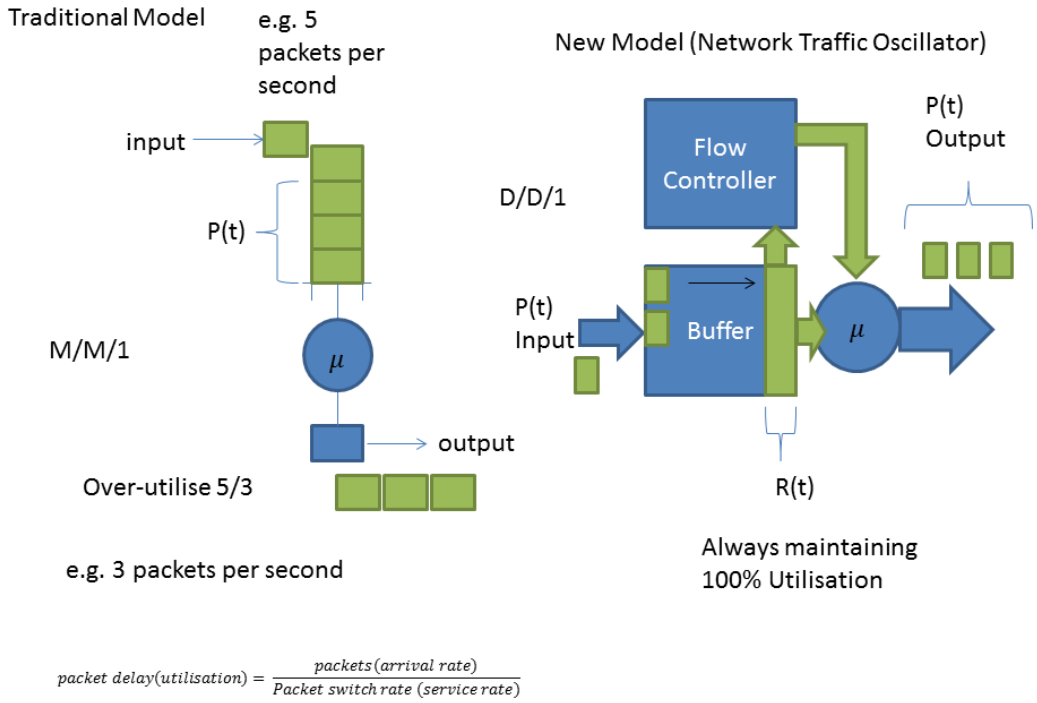
Theoretical maximum IP defragmentation (payload packet compression, the opposite of packet fragmentation) is achieved by using the maximum allowed data payload of an Ethernet frame (1500 bytes) [7, 45, 171]; Transport layer UDP and network layer IP overheads (28 bytes) reduces the maximum data payload to 1472 bytes (Table 5-1). All overhead measurements are taken from operational airfield radar. The standard r adar transmission payload field is 136 bits (even when payload is less than 136bits, p adding is introduced). The arrival rate of these transmissions is 5.7ms per packet. A t heoretical transmission of 86 packets can be fitted inside the maximum allowed Ether net frame(

Table 5-2), but this adds 490ms of waiting time to this batch packet, and sends 86 packets in batches to the control tower. The benefit of sending packets in batches is higher data transmission efficiency (less overhead). This transmission initially adds 490ms of waiting time, this type of IP defragmentation requires a highly reliable link, as this Ethernet frame carries all 86 packets into one frame. The total response time between packets is the payload of each packet (136 bits apart). IP fragmentation on the other hand reduces the initial waiting time of packets, but inherently increases buffering delay from packet overhead and higher delay variance from mismatched congestion management and retransmission [129]. IP fragmented packets have more overhead (47600 bits), compares to a 528 bit overhead with IP defragmentation (batched packets). Critical networking will optimise these parameters for the best of both services. Delay jitters are caused by mismatched transmission from an application to the network, namely packet storing, switching and routing using the prior art asynchronous Ethernet network, creating more arrival rate uncertainty in the control tower which is problematic for time critical data (air targets). Overhead data also increases response time of the transmission by adding more data to transmit.

This concept of transmission data compression has been known as Ethernet Frame Bursting for the Data-link layer, but this is done on an ad-hoc basis (randomly) and only benefits in terms of protocol efficiency as supposed to lower response time. A real deterministic Ethernet system cannot be achieved by simply Ethernet Frame Bursting alone - other networking concept improvements are also required to be made in conjunction.

5.3 The Network Traffic Oscillator (NTO) Implementation

Each safety critical application has a designated bandwidth requirement before transmitting into the network. The first task in implementing an NTO design is to identify the direct requirement of the real time application requirement. Packets in NTO are sorted by application and their payload per packet (R) before being transmitted into the queue (Figure 5.2). The payload is then transmitted into the safety critical network for low latency transmission.



$$packet\ delay(utilisation) = \frac{packets(arrival\ rate)}{Packet\ switch\ rate\ (service\ rate)}$$

Figure 5.2 Discrete NTO implementation using the Server Client Perspective

The theoretical NTO comprises of a buffer (which stores all application packets into one long payload) and a flow controller (which breaks down the payload into timeslots - packet transmission in batches). The server client model treats the buffer as the

packet queue and the flow controller as a server with acceleration packet transfer management (packets per second squared).

Table 5-1 The breakdown of overheads inside a safety critical packet transmission

Protocol	Overhead Type	Data Size (bits)	Total (bits)	
UDP	Source Port Number	16	64	
	Destination Port Number	16		
	Length	16		
	Check Sum	16		
IP	Protocol Type	16	160	
	Version Number	8		
	Service Code	8		
	Total Length	16		
	Identification	16		
	Time to Live	8		
	Transport Protocol	8		
	Header Checksum	16		
	IP Address V4 Source	32		
	IP Address V4 Destination	32		
Ethernet	Preamble	62	208	576
	Start Frame Delimiter	2		
	Destination Address	48		
	Source Address	48		
	Length (type 2)	16		
	Padding (max)	368		
	Frame Check Sequence	32		
	Extension[1]	3584		
	Total Overhead for Ethernet (Min / Max)		432	576

[1] The extension depends on the physical technology used (not including in table calculations)

Table 5-2 Maximum overhead compression

	Data(bits)
Maximum Ethernet frame size using IP/UDP	12208
Minimum total overhead for Ethernet frames	432
Space left	11776
Maximum ASTERIX Cat 34 payload size	136
Complete payloads in space left	86

The theoretical NTO treats the flow controller with a feedback control to increase and decrease service rate depending on the incoming packet. In the simplest implementation form, the theoretical NTO calculation can be incorporated into the server. It should be noted that this is not a pure aggregation system in the time division multiplexing (TDM) paradigm. The buffered payload is divided into packets in an adaptable fashion and transmitted at a rate that removes network congestion. The system delivers fixed packet time intervals regardless of the size of the payload per packet or the number of packets per second. The total time interval does not exceed the maximum application time delay specification. A shortened overhead is added after the NTO to give direction on networking; this includes just the sequence packet number.

5.3.1 NTO operation and principle

At the input to the device, the payload per packet is R_{IN} and at its output, this becomes R_{OUT}. The difference arises from the action of the flow controller (R_F) in proportion to the rate of change of output packets per second. An alternate NTO perspective can be seen in the instance of one fixed packet/payload in the system, and only the packet

time spent in the NTO has been altered. The different packet arrival rate of can be independent to the operation of the NTO, as long as the queue doesn't exceed zero packet/payload within service time (service rate inverse). This alternative perspective helps interchange between continuous NTO modelling and discrete NTO Digital Signal Processing (DSP). The NTO frequency is selected by the ideal application response time in one cycle of payload transmission.

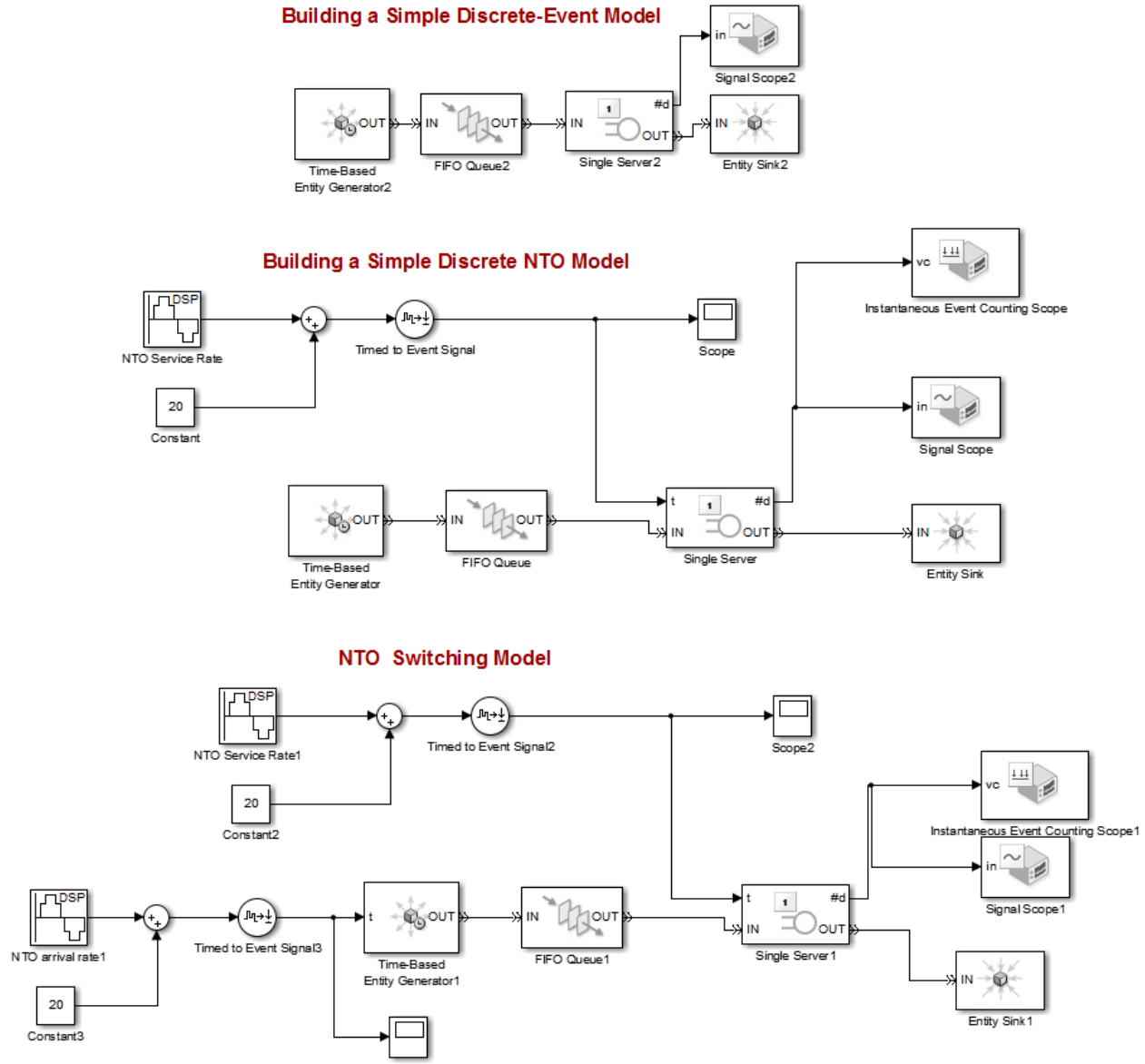


Figure 5.3 Discrete Simulation Model of the NTO in three stages a) consist of the basic queue server model, b) shows the modification of the queue server model using NTO. c) Using NTO serving and queuing model for networking

The service rate follows the zero order hold transformation from the Laplace NTO domain in Chapter 3. This NTO is designed predominantly to tackle the random congestion and buffer delay issues in the network layer. The NTO discrete simulation operations do not require a feedback loop from the queue to the server, the NTO buffer and flow controller loop model are used to observe the behaviour change between payload per packet and packet per second. In the discrete NTO model, this is modelled with respect to payload per second, and the result is created from the perspective of a single payload. Service rates are increased and decreased to remove link congestion and buffer overflow inside the network. The service rate is designed within these confines and the simulation demonstrates the discrete NTO in three stages (Figure 5.3). First stage shows no link between theoretical NTO buffer operation to the packet queue (Figure 5.3a). There is no difference whether there are more payloads stored inside the infinite FIFO queue or not. Second NTO diagram shows the operating conditions of the discrete packet event analysis (arrival rate and service rate) (Figure 5.3b) with random arrival rate payload. Third stage shows the response time example of one application as it get to the receiver with the critical network switching principle (Figure 5.3c). The NTO parameter has been calculated in pre-requisites to the application requirement and response time window.

The traffic characteristic result is displayed in Figure 5.4. A basic server queue traffic model processes packet/payload in a one to one ratio. The server time is the traffic shaper of an incoming packet and this has a regular time interval related to the service rate. When the queue is empty there will be additional increases in packet/payload time interval, and this affects the random arrival rate in other nodes of the network (Figure 5.4a). The NTO only operates when there are enough payloads in the queue for one cycle traffic oscillation. The packet/payload time interval increases

and decreases exponentially, this helps packet traffic to become deterministic, the exponential increase and decrease in service time helps to establish the buffering and flowing ratio and switching/routing management for ideal queue size and service rate management (Figure 5.4c).

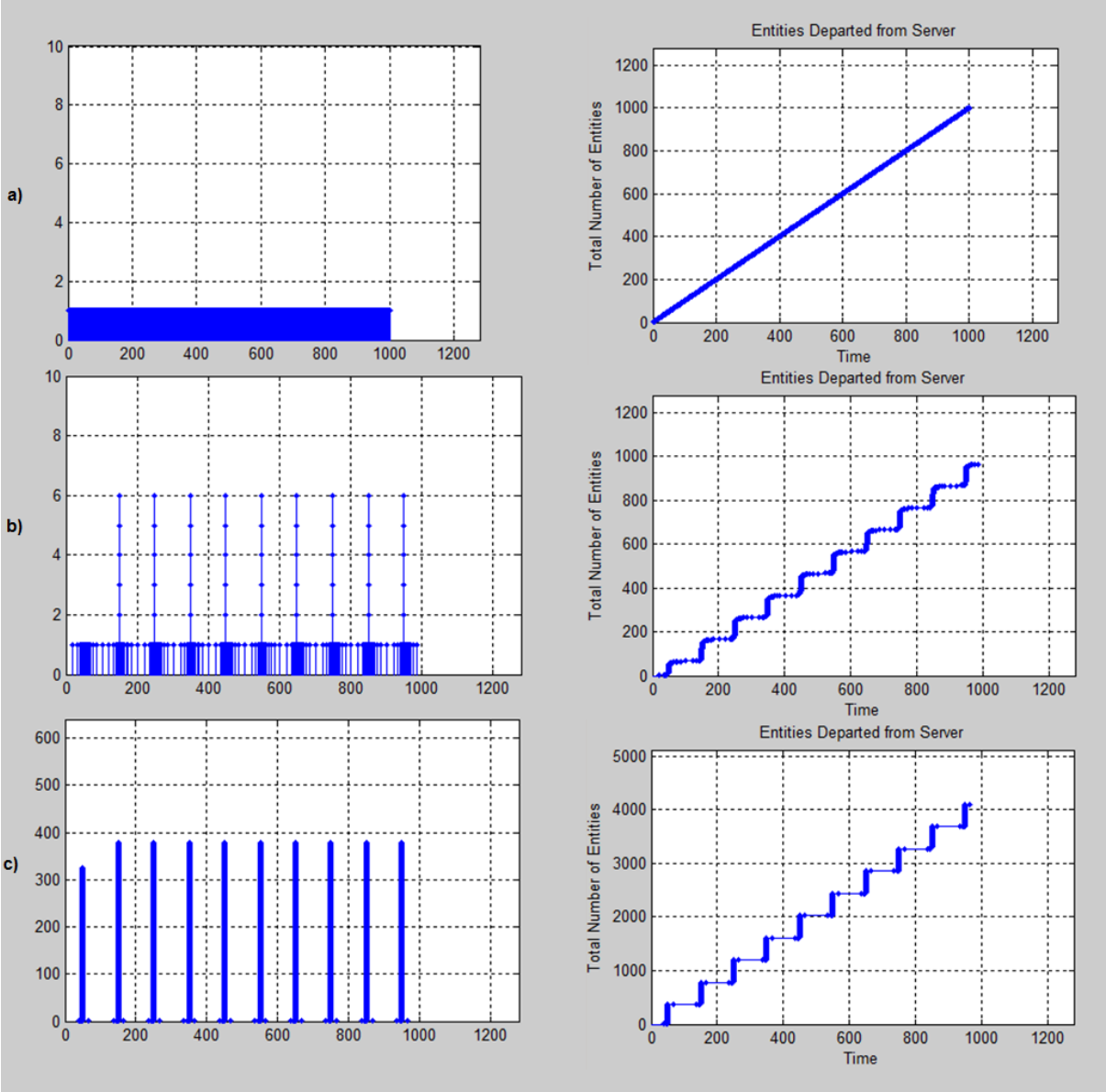


Figure 5.4 Shows the instantaneous entities (packet/payload) response on the left and cumulative results in the right. A) is the server client model without NTO and fixed packet payload size simulated per incidents, b) is the product of filling 1000 packets/payloads in a FIFO queue using NTO, packets are arriving simultaneous with a graduate increase and decrease of exponential service time, with the peak of 6 packet/payload arriving simultaneously. C) Is the result for NTO packet arriving rate with a matching NTO service rate, packet instances are arriving strictly at their designated time window

5.4 The NTO

The function of the NTO is to re-organise application traffic to better suit the network link capacity and it has two objectives. The first is to fix packet transmission in a predictable pattern for effective networking and the second is to bind the relationship between payload per packet (R) and packets per second (P). These can interchange depending on the network perspective. The NTO also produces a continuous sinusoidal payload per second model for network resource management. This allows a much better link capacity division amongst applications. The link capacity service rate uses network application payload per second to reduce the minimum delay in a network. All application payloads have a fixed response time of one cycle of the application payload oscillation. The end application receiver will combine one cycle of NTO payload per second to recover the arrival rate specified by the application design. Payload distance is then later used for network routing decision calculation. Other non-NTO applications can also access the link capacity by transmitting after all NTO applications have transmitted. The separation between NTO and non-NTO applications is the difference between a continuous transmission model and a discrete application packet transmission management. A continuous application transmission is a predictable transmission pattern (sinusoid) exchange of payload packet that happens for a long duration, while discrete communication generally happens in impulses. A series of large packet payload transmission impulses disrupts the balance between link usage and queue management. The link utilisation problem in the introduction section is an example of this additional network condition problem. The NTO uses application payload to confirm the exchange of incoming packets per second to payload per packet, and determine the true application payload per second

usage and the link utilisation service rate required. This task is handled by the NTO flow controller.

The NTO has two critical model components, a buffer and a flow controller. At the input to the device, the payload per packet is R_{IN} and at its output, this becomes R_{OUT}. The difference arises from the action of the flow controller (R_F) in proportion to the rate of change of output packets per second.

$$R_{OUT} = R_{IN} - R_F$$

Equation 5-6

Without the device, there is a buffer that accepts an input stream of pin packets per second and divides it into packets of size B

$$R_{IN} = R_B = \frac{1}{B}\int P_{IN}(t)dt$$

Equation 5-7

Using the NTO:

$$R_{OUT} = \frac{1}{B}\int P_{OUT}(t)dt$$

Equation 5-8

Now, for the flow controller:

$$R_F = F\frac{dP_{OUT}(t)}{dt}$$

Equation 5-9

So from (Equation 5-3 -6) the Laplace domain transfer function of the NTO may be obtained by recognising the resonant frequency $\omega_n^2 = (BF)^{-1}$:

$$H_{\mathrm{NTO}}(s) = \frac{\widetilde{\boldsymbol{P}}_{\mathbf{OUT}}(\boldsymbol{s})}{\widetilde{\boldsymbol{P}}_{\mathbf{IN}}(\boldsymbol{s})} = \frac{\omega_n^2}{s^2 + \omega_n^2}$$

Equation 5-10

The resonant frequency is low compared to the rate of packet arrival at the device and so the response observed will be that to a step input of size $\bar{P}$, the mean arrival rate in packets per second, thus:

$$P_{\mathrm{OUT}} = \bar{\boldsymbol{P}}\{\mathbf{1} - \mathbf{cos}(\boldsymbol{\omega_n t})\}$$

Equation 5-11

Thus, the buffer collects a multitude of frames in a period of instances into a longer payload per packet *R*. The flow controller divides a long payload per packet into multiple packets per second depending on the network workload. The directly opposing nature of the functions of these two components creates traffic oscillation patterns within packet transmission. The buffer factor *B* is the collection of frames and measures the level of traffic load in the network, while the flow controller increases or decreases the frame flow rate *F* based on the traffic load. These varying transmission patterns create a critical time response window between workload (frame size) and channel division management (frames per second). Oscillating traffic is managed within network switches and physical packet queuing buffers to deliver deterministic arrival rate transmission.

5.4.1 Critical Networking Switching and Routing

NTO success is to manage packet queuing effectively so that the incoming packets will never exceed the buffer size and cause overflow. The flow controller dynamically allocates link capacity bandwidth to cater for any sudden change of all incoming packet payloads in the queue. The NTO end result forms a fixed application link utilisation, creating deterministic payload per second (sinusoidal) traffic. The NTO oscillates the number of packet output and the application link utilisation is always 100% (the ratio of packet input and output is the same). An application link utilisation is different from the physical link capacity, which hosts the maximum traffic capacity, while other application link utilisation (their relative payload distance) is a subdivision of the physical link capacity. The NTO creates a Frequency Division Multiplex (FDM) scheme for each application to be subdivided. These application frequencies allow a critical networking switch to produce an individual networking scheme for improving connectivity (both payload size and arrival rate is known). The traditional IP network model [159] (Figure 5.4) regards traffic management with statistical properties, assuming the fact that packet congestions are only temporary. Arriving packets (Pin) and a service rate are treated as random. In reality, randomly arriving packets are caused by unknown application requirements, and random service time is related to the payload per packet (R) of each packet. This model includes a buffer in the middle (storing congested packets), which is often a first come first serve one. Traffic congestion is caused by mismanagement between the incoming packets flowing in and the transmitting packets flowing out. When there are more packets flowing in (packet arrival rate) than flowing out l(service rate), substantial network traffic congestion results (often exponential delay). The payload size per packet is the factor

that decreases the service rate of a link. The traffic congestion problem can be expressed with the Erlang C Traffic formula (Traffic Trunking problem), which expresses both the packet arrival rate (independently random) and the payload service rate (also independently random) as a probability of waiting (congestion delay), and serving (buffering delay) or blocked (packet dropped from finite queuing).

5.5 Critical Networking Switch

The concept of a Critical Networking Switch (inspired by time-space switching) is to re-organise network traffic with the minimum payload distance. From the transmitter packet arrival rate and the expected packet receiver arrival rate, as long as the total payload per packet to the receiver is also known, the Critical Networking switch can handle any discrepancy between the mismanagement of payload distance from the source to the switch, and from the switch to the destination. The Critical Networking switch can route traffic effectively by deciding the minimum payload distance from the source to the destination either by requesting more or less payload per packet (service rate) given the constraint of the link capacity in payload per second. Although, the NTO deterministic packet transmission rate is ideal for Critical Networking, as any continuous congestion delay can be compensated by a single phase shift from packet per second to payload per packet (alternating between buffering and flowing). Critical Networking traffic can also handle each networking incident as a discrete event (NTO continuous model is easier), but rapid processing power is required to respond to any unexpected change. A Critical Networking Switch would just maintain the NTO requirement, as it has already been re-organising traffic suitable for application, link

capacity and receiver transmission specification. A NTO payload distance would be the same (Same number of packet per second from the source as it is to the destination (Figure 5.5).

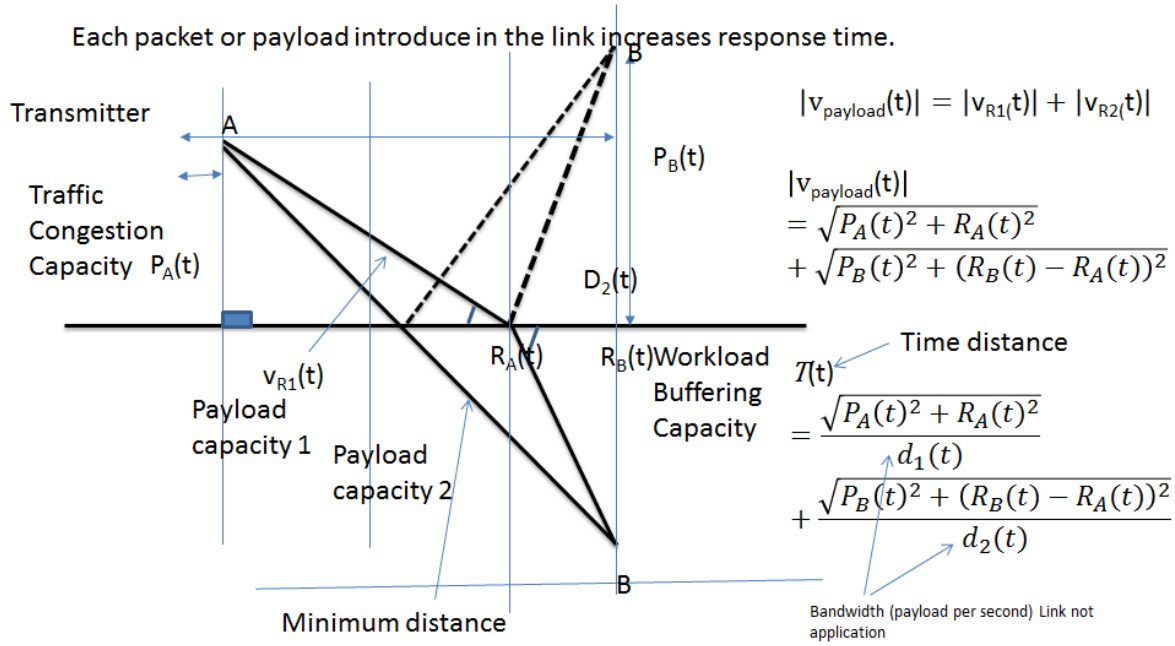


Figure 5.5 this is an example of a Critical Networking Minimum distance calculation. The minimum distance is always the same distance from the source (A) to the switch (origin) as it is from the switch to the destination (B). Similarly the best time response can also be measured by considering the reduction of payload per second rate to their respective distance

Other non-NTO application traffic could possibly send packets at a random packet arrival rate to the switch (labelled as PA) and with a random payload per packet rate. The total payload per packet and packet per second expected from the receiver is the area of the two parameters (D) (labelled as PB). The function of a Critical Networking Switch is to organise a traffic routing plan to reduce the delay between the receiver and the transmitter, even when there is a reduced payload per second service rate (link capacity) from the transmitter to the switch and from the switch to the receiver.

The reduced service rate is the difference between application payload per second requirement (application service rate) and the maximum link capacity (maximum service rate). The extra congestion delay can be worked out by the designated application payload per second (DA) and the link reduce payload per second rate (d1). The result is the increase of response time estimate between the distances. This deterministic congestion delay can also be used to decide whether to wait for more payloads per packet from the transmitter (increasing payload distance from PA to the switch) or to convert more payloads per packet to maximise the change in link capacity payload per second rate (d1) to the receiver (increasing payload distance from the switch to PB).

An NTO based network requires strict network resource management. Each network link must update its link capacity and reflect its information before assigning transmission. Dynamic network link capacity updating is crucial to maintain the Quality of Service amongst other real time critical applications (based on input measurement rather than another network management feedback protocol). Rather than using overhead (packet network information) to direct network traffic, flow can be directed using application frequency analysis. Frame overhead not only increases buffering delay by encumbering each frame with a larger payload, but it also restricts the level of freedom for network switching to manage traffic. Buffering delay is caused by large payload frames and congests the network by uncertain frame rates.

Using the NTO, the transmission of payloads with a fixed payload per second, D(t), by application protocols guarantees the oscillatory periodic transmission rates of both frames P(t) and frame sizes R(t). The two quantities P(t) and R(t) are ninety degrees out of phase using the NTO because of their sinusoidal nature and the relationship between them. The NTO parameters combine to deliver a deterministic payload per

second (33). Thus a switching arrangement is achieved to remove network delay completely (the delay is the measurement between two payload transmissions). The transmission rate, D(*t*), is maintained consistently in a network to ensure the lowest minimum distance between the source and the destination.

When a continuous transmission uses a large payload per packet resource, it should also use less of the packet per second resource. These two parameters, R(*t*) and P(*t*), create the payload distance of the link. To illustrate the fundamental concept, transmission between nodes N_A and N_B via an intermediate node N_C is considered as shown in Figure 5.5. N_A has the (packet, payload) coordinates (P_A, R_A) and N_B the coordinates (P_B, R_B). That is the number of packets generated in one second by A is P_A with a payload per packet R_A. It is thus possible to define a payload distance for link AC by the square root of the sum of the squares of P_A and R_A. This may be converted to a time T_{AC} by dividing by the payload per second value for the link N_A-N_C, which is denoted by d_1:

$$T_{AC} = \frac{\sqrt{P_A^2 + R_A^2}}{d_1}$$

Equation 5-12

Considering the transmission from N_C-N_B, a similar argument may be made to give a time over the link BC, with payload per second d_2, of:

$$T_{CB} = \frac{\sqrt{P_B^2 + (R_B - R_A)^2}}{d_2}$$

Equation 5-13

The two source rates P_A and P_B are fixed as is the total payload per packet at N_B. Therefore, the time is minimized by finding the optimum value of R_A by differentiating the total time $T_{AC}+T_{CB}$ with respect to R_A. .

Critical Networking encourages *application* payload per second to be the same as the *link capacity* payload per second transmission rate. This method allows the network switching process the freedom to delegate other link resources for other real time critical applications and transmit non-critical applications when the network becomes available.

The router setup can mirror that agreed by the NTO to produce a fixed time interval in a network, rather than variable uncertain congestion and buffer delay. This also removes the need for packet retransmission and packet timeout due to the fixed flow rate, eliminating packet loss from exceeding the critical time window. The timeless parameter *payload per packet* (R) is the main concept for network traffic oscillation and network workload management, and it is only increased in size when multiple packets arrive simultaneously (within the hold time control). The buffer regulates payload to packet; when packets are merged, an effective application payload is released as payload per second (D). The buffer size dictates the pace of each individual application's payload per packet and its payload per packet operates with the flow controller. The function of the latter is to evenly distribute the available physical medium bandwidth into packets (time slots) but depending on the payload, it should also increase the packet distribution based on payload. The two aspects of payload per packet and packets per second form the basis of the NTO time vector metric when one time slot of transmission is considered that will be sent over a link with payload capacity d_L

By careful choice of the P and R values for each transmission in the network (packets are separated into different queues based on their traffic load), it is possible to minimise this metric and match *application* payload per second to the *link capacity* payload per second. Correct matching removes random queuing delays and enables deterministic performance to be achieved. Random P and R may appear in the router. Traditionally EIGRP has been utilised to manage packets on an individual basis). The NTO promotes a separate queue for each type of packet, with the payload per packet R used to achieve the separation. The expected delay is calculated by measuring the magnitude of the payload per second of the packet/payload. The expression (Equation 5-16) is compared to the lower expected service rate (d1) also in payload per second per server queue. When the router is placed in a mesh network, the total delay expected of the packet depends on its P and R values as well as the arrival rates of P and R at the sender. In Figure 5.6, a mesh network of two nodes with a router in the middle is considered. The sender (node A) can transmit a random number of packets per second (P) to the router with a random payload per packet sizes (R). The router has the ability to reduce or increase the payload size of the packet to improve the response time to node B. The total payload distance can be worked out by calculating the payload distance from node A to the router and from the router to node B. The time delay of this transmission depends on the service rates of the two links (A to router, router to B), this is labelled D1 and D2. The service time (buffering delay) is determined via the inverse of the service rate (Figure 5.5).

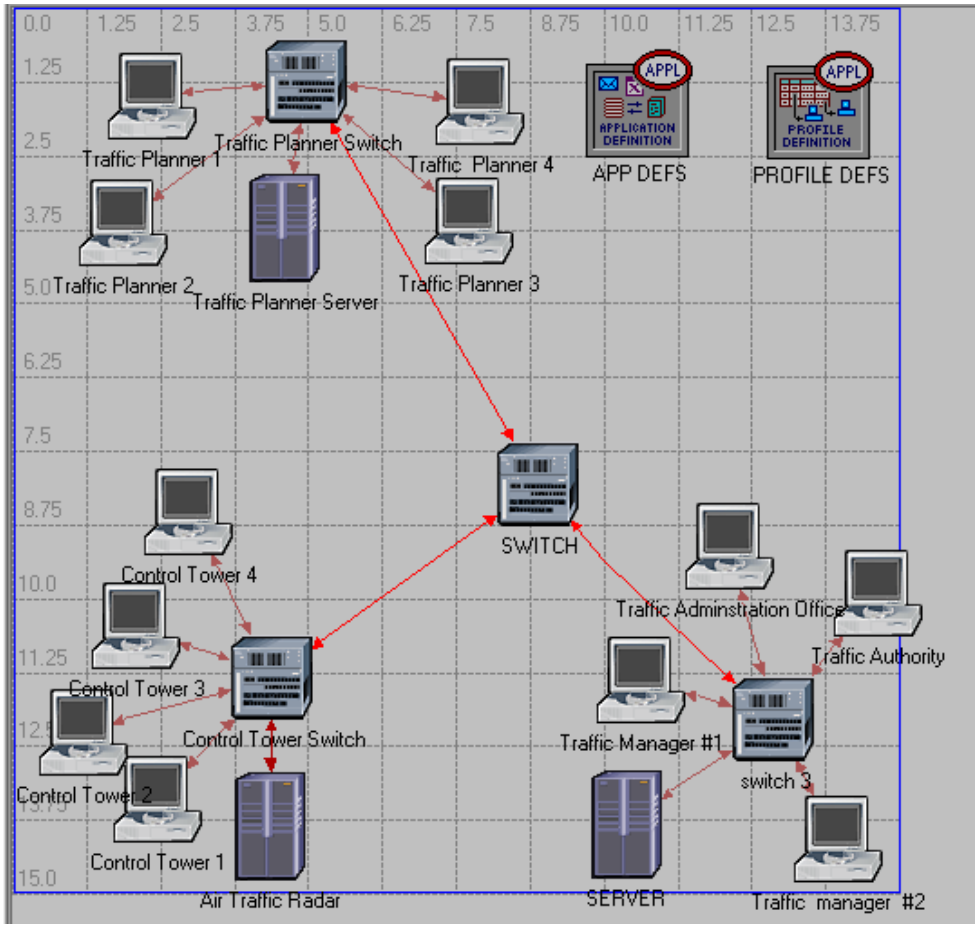


Figure 5.6 Real time critical Ethernet network simulation designed to mimic the air traffic SWIM infrastructure in an airport

The lowest service time is the minimum service time of the payload size (Equation 5-14). The true time distance is worked out by the similar relationship of a sine function (Equation 5-15). The minimum delay is when the link transmission delays match; this yields a "Snell's law" using payload per second (service rate) (Equation 5-16).

$$T_{NTO} = T'(t) = \frac{R_A(t)}{D_1(t)\sqrt{P_A(t)^2 + R_A(t)^2}} - \frac{R_B(t) - R_A(t)}{D_2(t)\sqrt{P_B(t)^2 + (R_B(t) - R_A(t))^2}}$$

Equation 5-14

$$sin(\omega_1 t) = \frac{R_A(t)}{\sqrt{P_A(t)^2 + R_A(t)^2}} \text{ and } sin(\omega_2 t) = \frac{R_B(t) - R_A(t)}{\sqrt{P_B(t)^2 + (R_B(t) - R_A(t))^2}}$$

Equation 5-15

At $T'(t)=0$

$$\frac{sin(\omega_1 t)}{D_1(t)} = \frac{sin(\omega_2 t)}{D_2(t)}$$

Equation 5-16

An oscillating traffic model would have an angular payload velocity, and using this property, the NTO traffic only requires angular phase shift by rearranging the sine function on the left hand side with the different of the link payload per second ratio (Figure 5.7).

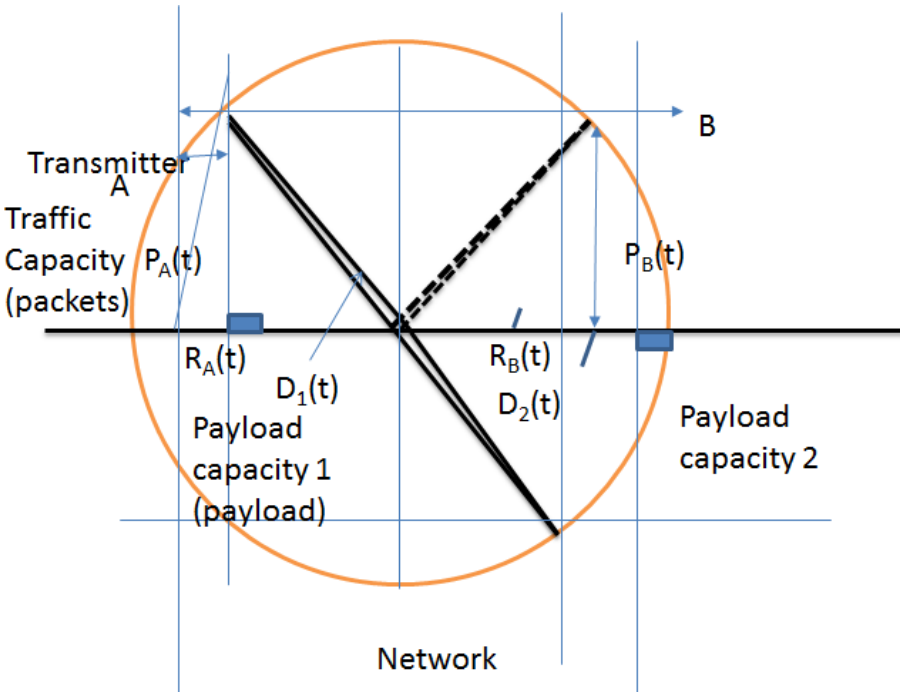


Figure 5.7 Oscillating Traffic can easily compensates the different in transmission rate (payload per second service rate) of the two link capacity

5.6 Zero order hold (ZOH) filter

The final component is a critical zero order hold (ZOH) *z*-filter, which enables the Critical Networking Switch to predict the NTO traffic (which has been designed in the Laplace domain) to interface with a digital transmission system. This naturally enables the Critical Networking Switch to be in time-domain filter form for signal processing in each network connection. An ideal sampling time (T) corresponds to the trip timer of each connection (Switch/Router/Hub). This reduces unnecessary queuing during the transit terminal (Switch/Router/Hub) and ultimately decreases application response time. The resulting form of the NTO ZOH is given by:

$$H_{\mathrm{NTO}}(z) = (1 - z^{-1})\mathbb{Z}\left\{\mathcal{L}^{-1}\left[\frac{H_{\mathrm{NTO}}(s)}{s}\right]\right\}$$

$$\Rightarrow H_{\mathrm{NTO}}(z) = \frac{(z^2 - 1)(1 - \cos(\omega T))}{(z^2 - 2z\cos(\omega T) + 1)}$$

Equation 5-17

Often the underlying physical switching technology scheme is fixed but application transmissions dynamically change their requirements, making deterministic transmission impossible. The combination of the sizes of the buffer and the flow-rate thus form a transmission resonant frequency response, offering each transmission a deterministic arrival rate for critical physical bandwidth allocations rather than a random one from the application layer.

5.7 Results

There are two types of transmission in an Ethernet system, continuous and discrete. The former is normally large payload transmission over a long period of time, whilst the latter is generally a smaller payload over a shorter period of time. The NTO enables the time-critical accommodation of continuous transmissions such as live radar feeds needing a specified transmission window time, which were treated as a series of discrete communications in the past. The NTO allows continuous real time transmission to be more flexible (interchanging between P(*t*) and R(*t*) transmission) for link capacity management, while operating within the time response design of the application (one over the payload per second). NTO continuous real time transmission fixes the link capacity usage of the network. As a real time application transmission is oscillating, other real time applications can use any other available link capacity resource. Other continuous application transmissions are encouraged to transmit data out of phase with all existing continuous transmissions; this reduces the knock-on effects of increasing payload distance from payload per packet, R(*t*) and packet per second, (P(*t*)).

In Figure 5.8, the results show that continuous real time application transmissions can co-exist with normal commercial applications such as e-mail, database access and server access created by user terminals, which do not require NTOs as they can use frame overhead to direct their transmissions. Link resources are allocated to NTO transmissions first and thus the response time of these is deterministic as the link always allocates the same resources to these communications. The clear result is that servers 1, 2 and 3 deliver deterministic service to the real time applications despite the presence of variable traffic loads in the network, including random data bursts from

the background applications. This demonstrates that Ethernet has the capability to deliver the required service to the radar traffic in an airport scenario without dedicated links.

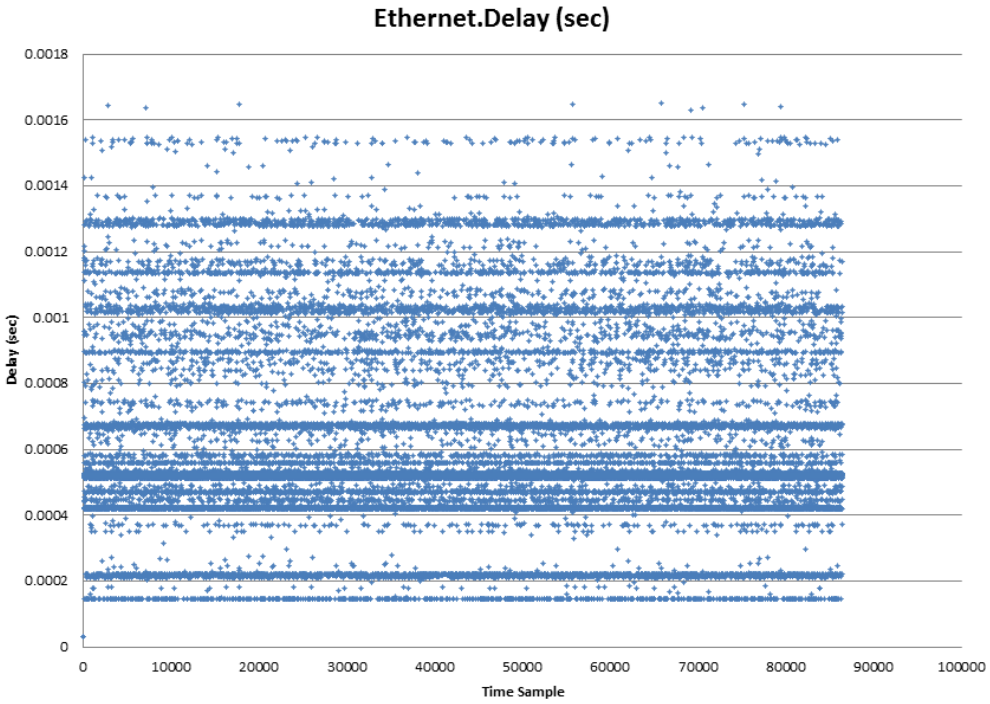


Figure 5.8 The Air Traffic Communication inside the simulated network scenario

Although critical networking could in principle be achieved without an NTO, since it concerns maintaining the minimum payload distances in a link, the level of optimization required would need advanced knowledge of each real time continuous transmission behavior. This is impossible when any applications can use the link by adding in the correct frame overheads.

Often sudden discrete communications that are non-time critical unintentionally offset the payload distance (P(*t*) and R(*t*)) of time critical communications. Priority queuing [172] is difficult to achieve given the unknown frame arrival rate of each real time

critical communication session. Links that have a large bandwidth (in bps) can transmit payloads faster, however increasing the application payload, D(*t*), per second due to this extra link capacity also creates application payload distortion effects. This is not noticeable when the application transmits discretely (not treated by NTO), but noticeable when the continuous transmission is managed by the NTO. The level of distortion is directly proportion to the different payloads per second between the two link bandwidth technologies. This distortion can be repaired by adding phase shifts to the NTO frame transmission and to the payload per frame. Link capacity management can be easily simplified by just maintaining the same payload per second regardless of the underlying switching bandwidth.

5.8 Conclusions

As the size of a network increases, resource planning is increasingly difficult when the network relies solely on overheads to direct traffic. Worse still, overheads also cause larger payloads amongst frame transmissions since bigger networks naturally need bigger frame overheads i.e. larger addresses, greater frame padding, more information for additional multilayer network services [7] and support protocols for frame diagnostics. Such overheads introduce both buffering problems and also additional failures in frame transmissions via overhead errors. Frame overhead is useful given large dedicated bandwidths and link capacities but offers diminishing returns when the bandwidth and link capacity are low. Although each frame can be identified by using a frame capturing tool, this level of frame quality assurance only measures the quality of the information presented; it has nothing to say concerning the reason for the delay of the frame. It is infeasible to operate real time critical

communications in an environment where they may be delayed by discrete, non-time critical commercial messages, which should give priority to the real time needs. Here, the utility of the NTO concept to address this problem and offer real time deterministic performance in an Ethernet network also carrying other non-real time applications has been illustrated. The use of payload distance to quantify the performance is facilitated by the NTO, which delivers a controlled traffic stream into the network. The real time traffic is shielded from the effects of users sending large non-time critical payloads by the NTO. By managing the transmission rate at the input to the network, flow management is also made simplified since the uncertainty in frame arrival times is removed reducing congestion. An uncertain arrival rate transmission discourages any effective advance routing planning. Network resource wastage such as low data utilization occurs because low payload frames are kept in a buffer even when the link capacity is perfectly able to handle them –this problem is removed using the NTO. In short, it has been shown that Ethernet can deliver deterministic service to critical real time applications without dedicated links and in the presence of random traffic from other applications.

Chapter 6 – Application and usage

The application usage for Network Traffic Oscillator (NTO) can be used in a long haul communication and in increasing communication security from cyber-attack in a network. Long haul communication is part of SESAR project to connect multiple airport sites together.

6.1 Example Application: Airfield Radar

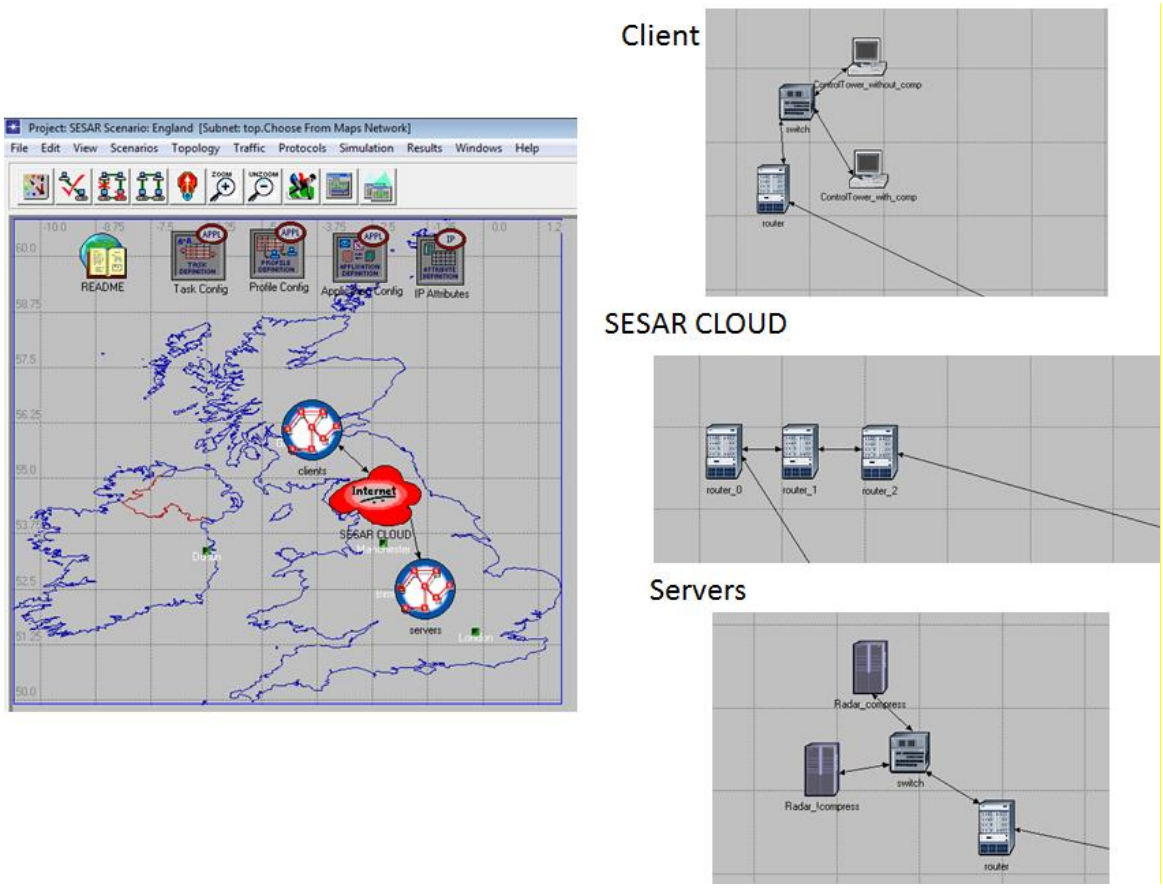


Figure 6.1 OPNET discrete network simulation topology

Having presented the framework for the NTO, we now present an example of its application using operational airfield radar data captured by Wireshark [173] . The case study concerns the communication between Glasgow and Coventry airfields, as described in section 3.2 (Figure 6.1). However, matching the strict critical communication time window set by the Glasgow control tower is challenging because although Coventry airfield radar immediately transmits the information it receives, there is only a limited network response to support this transmission, coupled with the physical technology restrictions of European Carrier (E1) and Synchronous Digital Hierarchy (SDH)[[171, 174].

The simulation assumed that the link from Coventry to the SESAR Cloud (Cloud computing [175]) and from SESAR Cloud to Glasgow are relativity high speed (high throughputs). This high speed link is using PPP (point-to-point protocol) SONET [174, 176]. The SESAR CLOUD is managed by three routers. The result shows that generally the compressed radar uses less data throughputs than the uncompressed radar, however the compressed radar has an increased response time to the client when compared to an uncompressed radar. The real benefit of using payload compression comes from when the connection link is of low data throughput E1 connection [177]. Not only has the compressed radar used lower bandwidth (data throughput), but the radar also has a faster response time from the client.

6.1.1 Scenario Details

In the past, the transmission of radar (a low but critical payload at periodic time intervals) would have been achieved with an E1 link. The distance from Coventry to

Glasgow is 408 km and so there is a trip time delay of approximately 2 ms over copper cables. The E1 physical switching technology has a data bandwidth of 2.048 Mbps, with 8 bit time slots, 32 time slots in a frame and 8000 frames/s. Thus, the 136 bit radar payload takes 17 time slots to complete and thus consumes a complete frame of 125 µs. Thus, when the radar transmits a packet every 5.7 ms, the total delay using E1 (radar plus protocol plus physical delay) is approximately 5.8 ms. This design requires (i) a dedicated communication link which is expensive with low utilisation; (ii) no overheads making it incompatible with networking; (iii) a low payload with fixed time intervals between transmissions.

Legacy radar equipment has a relatively low payload requirement but a reasonably high message arrival rate (low time intervals between transmissions) and thus needs a high bandwidth to maintain its target of providing rapid client-server response. Investigation of current airport practice found that radar data packets have a high overhead and the effective data transmission efficiency was low (23.6%)[29] . Although a dedicated high bandwidth link from Coventry to Glasgow with the fastest response time could cover all overhead cost, maintaining such a service would be relatively expensive. One possible solution is to remove unnecessary overhead by packet to frame compression combining multiple packets into one, creating a more effective transmission, which we term IP defragmentation. For time critical applications this is preferable to IP fragmentation [20, 128] which adds more overhead and produces a higher delay variance as a price for lowering the time interval between transmissions [129].

6.1.2 Results

To investigate the approach described above we first considered the performance of the NTO only. The stochastic radar communication traffic gathered in the field was fed into the NTO. The output formed a modulated sinusoidal output (Figure 6.2). The device not only produced a fixed payload of 3.5 kbit but also an optimal time period of 3 ms for this payload delivery system. This results in a more consistent transmission process with a bit rate of 32-35 kbps that is more reliable than transmitting the raw radar payload signal.

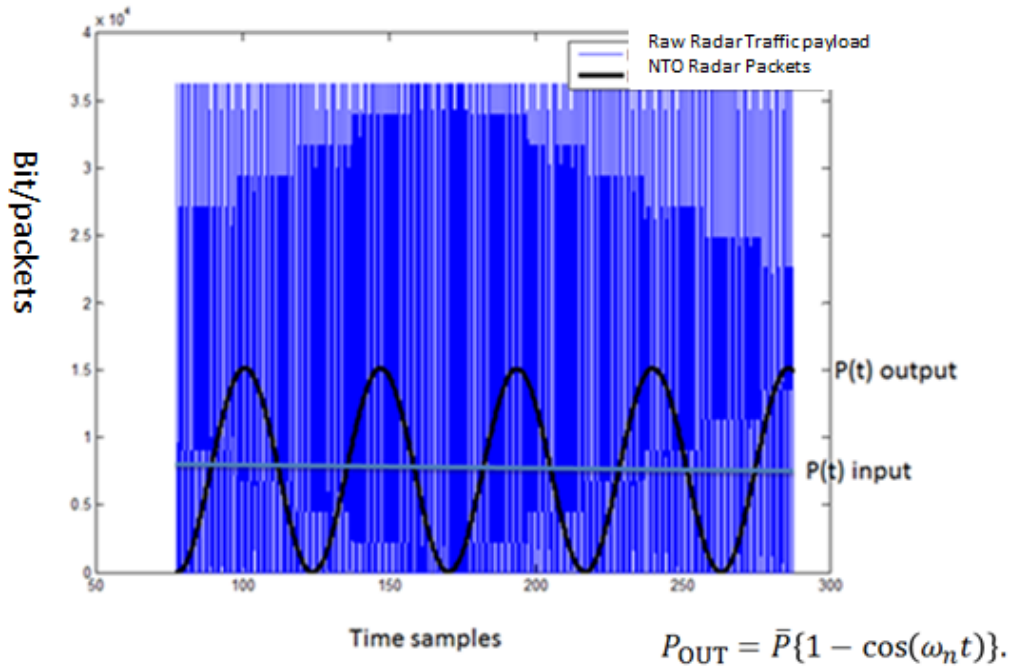


Figure 6.2 Raw Euro-Control Radar with ASTERIX Protocol being fed into a NTO

Next, an OPNET simulation was implemented for a Wide Area Network (WAN) connecting Coventry and Glasgow with three routers representing the SESAR SWIM cloud as shown in Figure 6.1. Two options for physical media were compared in the

simulation, SDH OC-3 and E1 using Point-to-point Ethernet (PPoE). The “server” end in Coventry includes two radar systems, one that implements a NTO and one that feeds raw data. Results are collected by two types of “client” in Glasgow, one for the critical network radar information and one for the raw radar peer-to-peer application data. Both server and client local networks employ 100BaseT Ethernet. The application transmission was either managed by the NTO including the ZOH with a 30 ms sampling time, or using peer-to-peer protocols without the NTO for comparison. The buffer size (B) was 86/136, and the flow rate (F) 86; Q_1 was 4.24 (pre-critical networked) and Q_2 was 1.073 (critical networked). The results of the simulation are presented on Figure 6.3 (payload size distribution) and Figure 6.4 (time response distribution).

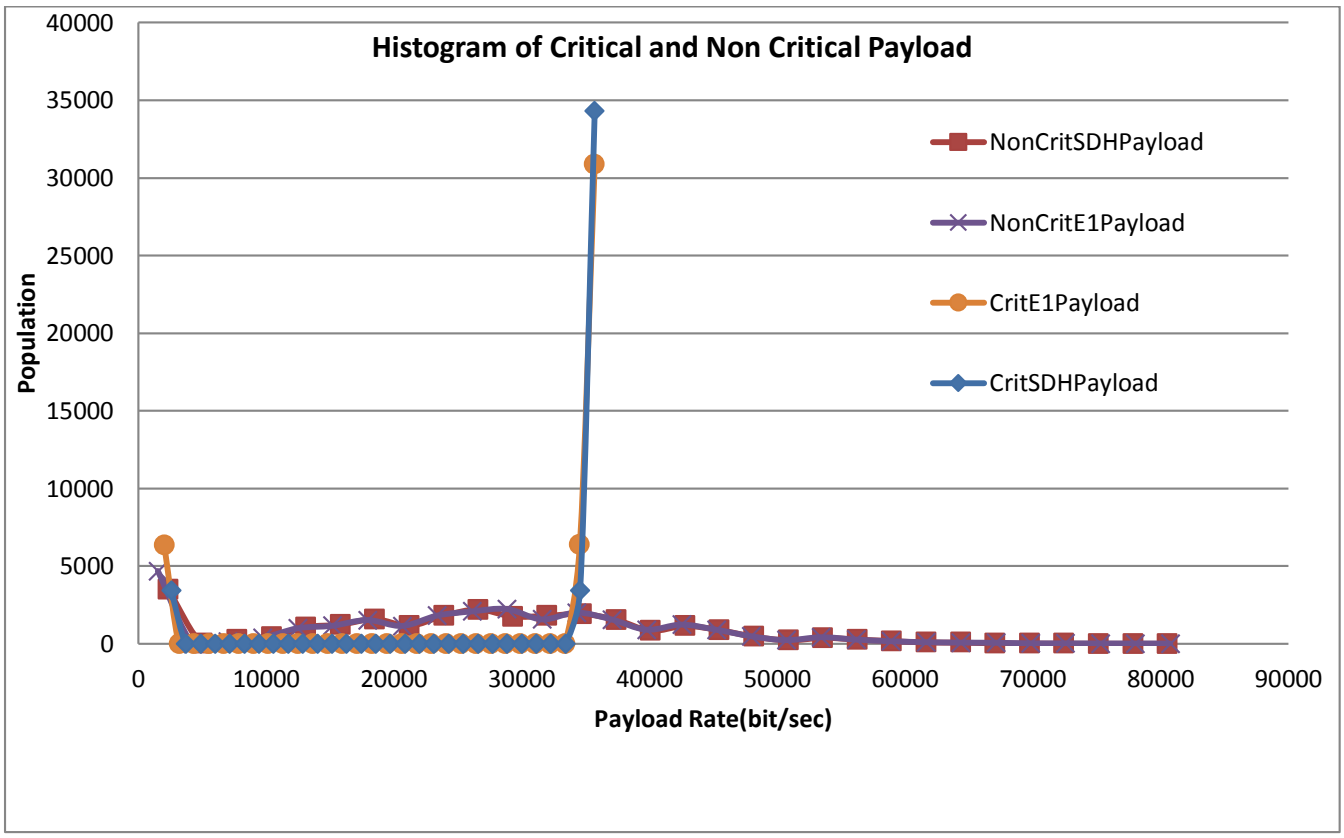


Figure 6.3 Payload distributions for critical and non-critical switched networking across E1 and SDH OC-3

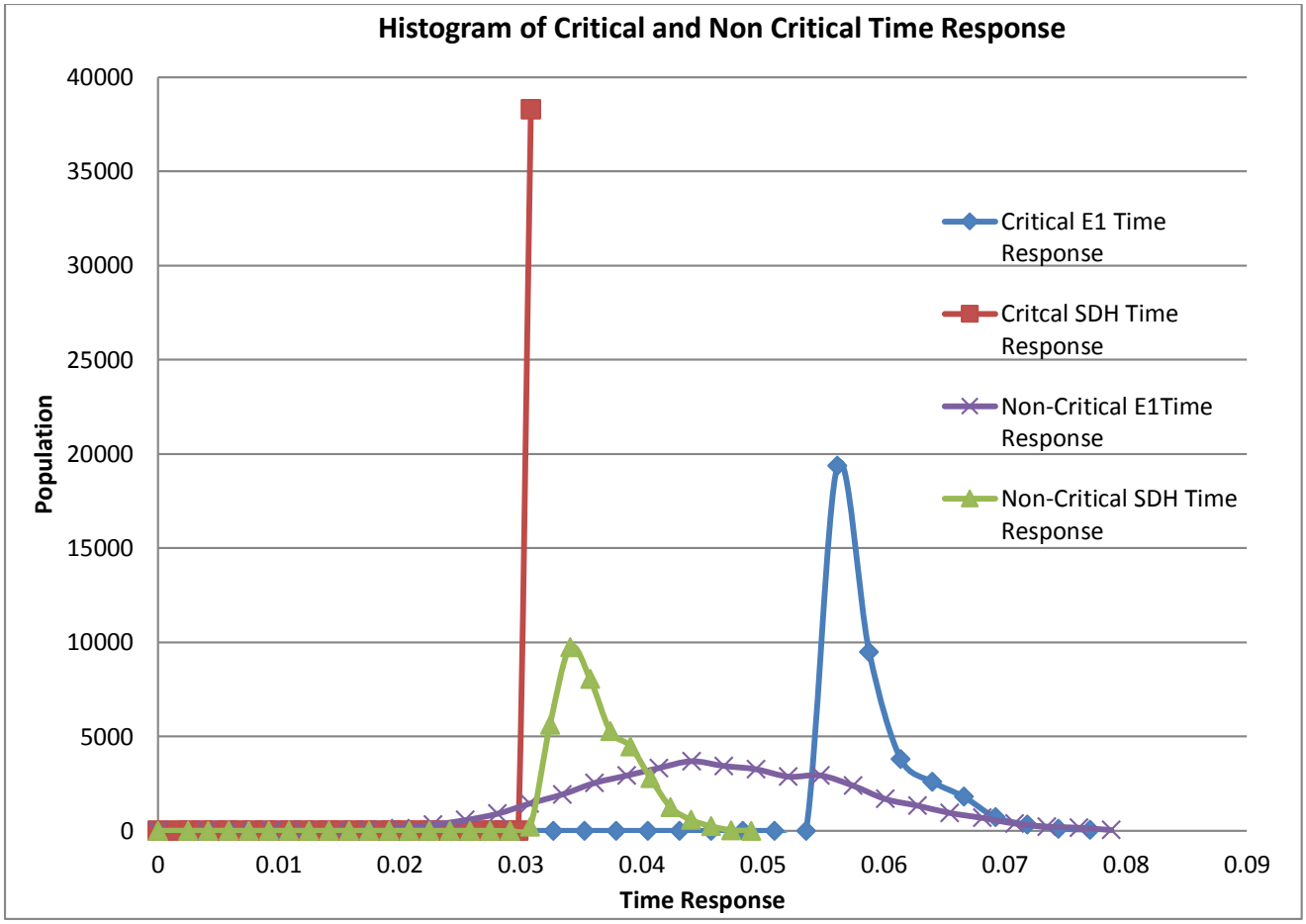


Figure 6.4 Transmission time distributions for critical and non-critical switched networking across E1 and SDH OC-3

6.1.3 Discussion

In Figure 6.4, it may be observed that the payload is constant for both E1 and OC-3 as expected when critical networking is employed. In contrast, the standard transmission scenario displays considerable variability arising from the stochastic nature of the arrivals being passed into the network rather than ameliorated by the NTO. For the arrival times, OC-3 reproduces the concentrated shape seen for the packet sizes when critically networked using all of the above technique developed in this thesis, whereas E1 has a tail in its response as a result of its lower speed and frame structure from over-congestion network. This is cause by miss-matching between service rate and

arrival rate mention in Chapter 4. Both produce results that are substantial improvements in the situations. NTO produces a tail in the response occurs and for Non-NTO, the time variance is extremely wide. In both cases each E1 incompletely transmitted packet has to be allocated to the next available time slots until the total expected packet is received. Small packets (with low payload) have faster response times in E1 but the overall packet group has a higher response time variance resulting in radar information being discarded because of wrong time stamp. Overall, critically networked packets improved the precision in targeting an expected response time, even when technology such as E1 requires considerably more time frames to achieve the same results as OC-3, payload and packets arrivals in orderly sequences for those error-free and time critical applications.

6.2 Network Layer Security and beyond

A transparent network allows cyber-attacks to be spotted. Cyber-attack concerns the infiltration of security networks and generally refers to methods for unauthorized network manipulation and control [178]. There are two types of attacks, technical and social. The former focus on targeting the mechanical weakness of a network, these attacks include the targeting of firewalls, routers and nodes (end network devices). In contrast, the latter types of attack use social engineering to steal personal details from the victim, such as user name and password, and this attack is formally known as phishing. The only solution to prevent a social cyber-attack is personnel training. Here we consider the use of critical networking (by which we mean a philosophy for transmitting only the essential information (payload) to the receiver within the critical time frame) to neutralize cyber-attacks by unveiling its operation. A common technical

cyber-attack is the Denial of Service (DoS) attack [179], which disables access to a node, a port or a service by packet congestion. Before such an attack exists, the network hacker is required to develop a substantial knowledge about the type of network, its configuration and security.

Existing technologies such as firewalls and virtual local area networks (VLANs) can protect users from these forms of security intrusion when deployed correctly, but often these technologies have difficulty in covering all weakness in a network. Even when the network is fully protected, operating across network becomes highly inflexible when a new service is introduced into it. The NTO is introduced to renovate the current way of network application management.

One major weakness in security networks is its conformity Ethernet operation but not its network transmission operation. Most information can be gathered by reading the overhead of packets. The only way to prevent packet reading is by pre-emptive firewall port blocking, or elaborate virtual network planning, and this involves transparent network operation.

6.2.1 Firewall Protection

The OSI (i.e. ISO/IEC 7498-1) model provides an insightful explanation of device functionality in a network, to which firewall types conform. Commercial networks also operate in layers, each type of device in a network functioning in a different layer. In summary, network switches operate at layers 1 and 2 (physical and data-link), routers at layer 3 (network), and workstations at layer 7 (application), and there is a firewall type for each layer [180].

A firewall is the primary defense against technical cyber-attacks and can generally

limit the type of protocols, addresses and devices that may access the network. First generation firewalls limit access by packets, and are the most powerful in blocking traffic, but are highly inflexible to network changes due to the fixation on packet types. A generation 1 packet filter samples each packet to detect unauthorized packets in a layer 2 data stream. The second generation limits traffic by transport protocols (UDP/TCP). Generation two firewalls use stateful filters[181], which employs a three-way handshake to synchronize with a sender and a receiver, followed by approved layer 4 acknowledgements from the two parties. Each node in a network must also be approved before transmitting. Generally, a network administrator is required to manage every network transaction. Generation 1 and 2 firewalls are effective at blocking unauthorized access by bottom-up network layer filtering, but difficult to fully implement in a large network due to the consequent intensive micro-management. Generation 3 firewalls restrict network access by application type and are flexible with respect to network change but need insight for maximum security since applications may add and drop dynamically. Critical networking breaks down the seven layer OSI perspective into two layers (application and physical), and focuses on macro-management and network resource (physical bandwidth) protection by application protocols. Regulated application transmission provides clarity in network usage. The detection of unauthorized transmission is made possible due to the resulting improvement in network transparency allowing the creation of a hybrid firewall.

A cyber-attack has two phases, the first being packet sampling, which collects information about a network, and which often takes the form of malware to mine packets inside a device connected to a network. This is followed by transmission of data back to the cyber-attacker's computer for security assessment [182]. Anti-spyware scanners were invented to counter such operations, but often the information

has already leaked, unless there is pre-emptive intervention by the firewall. The second phase is the control and steering of an authorized network [183]. Cyber-attacks often involve congesting a port by overflowing a traffic buffer, overloading a node with packets, or injecting false protocol information. In this scenario, generation 3 firewall protection in this network has failed. Here, the network implements critical networking, so the detection of unauthorized access is more transparent than the hacker's attempt at network infiltration. A hybrid firewall is later imposed via traffic filters to selectively target and block unauthorized access. This hybrid firewall traffic filter is built into gateways, switches and routers, and the configuration is unknown to all end-devices at the application layer.

This hybrid firewall is only effective because of the rigorous packet assessment from the NTO. This includes checking for application payload and temporal conformities with unmatched transmission characteristics flagged as potential problems in the network.

6.2.2 Application to Physical Bandwidth Separation

Fixed periodic oscillating transmissions at regular intervals are better in busy networks because they have upper and lower bounds on their rate. In contrast, instant random application transmissions easily escalate to produce congestion, especially when application protocols are disconnected from network routing algorithm (problems such as broadcast storm [184]). Direct application packets do not necessary correlate with the actual required operational physical bandwidth. Network traffic management is achieved by introducing an intermediate stage for the oscillating payload to form a pattern to combat hidden transmission amongst other application transmissions. In

general, a small overhead factor is ideal for any effective information transaction. Overhead should not be introduced in the network oscillator to avoid steady state transmission resulting from the oscillator effectively becoming a step function. Applications can no longer be distinguished by frequency or payload magnitude. The buffer and flow rate size are tailored, dependent on the application demand and the available network physical which both specify the oscillator frequency. A large network buffer regulates burst communication by collecting stochastic packet transmissions into a large payload, whilst a small buffer produces low payload for reduced delay transmission. The flow rate, *F*, is responsible the oscillation rate of application bandwidth allocation. Low values are appropriate for low bandwidth transmission with low overheads whilst high values are suited to high volume payloads with relatively high input overheads.

Traffic overflow is prevented by appropriate spacing via the transmission fixed time interval (δt). The critical application bandwidth refers to the situation in which communications is via one large payload packet over the hold time. The physical payload per second remains unchanged since packet and payload rate are inversely proportion to each other. The physical payload per second is only affected by the rate of change of time intervals. Given that the time interval is universally known within the network, all applications have a deterministic payload per second transmission. A packet inspection system can separate application protocols by monitoring and filtering payload parameter with respect to time. In reality, critical networking must be engineered by the buffer-flow controller combination. A maximum sampling frequency is made using the smallest time interval suitable for

both the buffer and the flow controller. The construction of a Fast Fourier Transform (FFT) uses this default parameter to scale the total recorded dataset appropriately. Adaptive packet traffic filters are possible by the construction of Zero Order Hold (ZOH) digital filters from the transfer function $H_{NTO}(s)$ of the NTO (Equation 5-17) where T is the hold time. Although the idea of filtering per se is not uncommon within cyber-security [178], there has been to date no establishment of such an architecture for high accuracy traffic sampling, monitoring and blocking. Moreover, this method facilitates a simpler solution by utilizing traffic filters.

6.2.3 Automated Packet Inspection System

Network traffic monitoring for the network administrator is challenging [179], and becomes impossible in real time when there is a high volume of packet transmission all directions. Whether it is via packets, protocols or an application, a traffic monitoring system must be able to collectively distinguish between normal traffic behaviours and cyber-attacks. In contrast to the current packet transmission architecture, requiring real time traffic monitoring from the bottom three layers, critical networking only required the network administrator to observed application packets. Not only is this solution far less burdensome than keeping up to date with all activities from three layers, but also network traffic analysts can easily distinguish between types of applications and their normal traffic behaviour. Furthermore, the approach simplifies the refreshing process that facilitates tracking of the on-going upcoming traffic in a network. Privacy can still be maintained by presentation layer packet encryption/decryption. Packet inspection here is only concerned with the application

traffic relative to that expected as opposed to current inspection techniques of collecting and storing traffic packets. Thus, the NTO generates traffic patterns which are later used in this network inspection system for monitoring traffic. Transmission that drifts from this pattern is an early indication of failure network equipment or cyber-attack. An effective way of observing all application is by dividing all application transmissions using the Discrete Fourier transform, with each hold time being the refresh period for each analysis.

6.2.4 Hybrid Firewall

This Hybrid Firewall has the blocking power of generation one packet filters, the authentication process of generation 2 transport layer approaches, and keeps tabs on all application protocols, as per generation 3. This combination of firewall designs is made possible by the transparency introduced by the network oscillator. The proposed adaptive filtering method is possible by the rearrangement of network packet transmission. An ideal firewall is one that requires little micro-management, whilst maintaining easy macro-management and high security. By using the network oscillator we gain greater application transmission insight to offer tailored capabilities drawn from all three generations, without the need to increase in firewall design complexity. By monitoring, this self-adaptive filtering process is able to tune into the transmission pattern of a network application. Since a cogent firewall is only potent at the weakest network point in the network, the combination of firewall properties here offers a unified hybrid firewall model to every network device for enhanced security. Firewalls should coordinate and exchange their adaptive filtering model to each other, so that anomalies can be detected from transmission patterns [185].

6.2.5 Methodology

Sensitive networks are ones that contain and manage time-critical information across safety-critical premises. This study focuses on the prevention of cyber-attack in an airport, which contains information regarding air traffic control, weather, air traffic plans and radar. Historically, each of these applications has been maintained by individual equipment and local serial networks. However, as both the information volume and the network expanded to include more detail and third-party support, standard off-the-shelf equipment was made to allow more inter-connection and service in the airport. This rapid expansion in applications has created a highly problematic security scenario. In the simulation, information is held inside its respective virtual application server but in reality there is only one central physical SWIM [18] server with many ports for each application. The network is divided into application virtual networks whereas in the past, equipment has had its own individual network connection. Equipment instances with two application functions are still assigned to the same network configuration. Application virtual networking is made possible by the introducing network oscillation; separation of applications allows better traffic management for switches, routers and network administrator.

This demonstration includes the identification and blocking of a cyber-attack via three stages. The first includes the conversion of a single application, such as File Transfer Protocol (FTP), into a network oscillator. The device operational parameters (payload amplitude, frequency, time slot) are agreed by the SWIM architecture network. The second is the monitoring process linking the application performance and its real time payload-time sampling. The final stage is imposition of a zero permitted transmission

time slot and restricted payload size filter by the Hybrid Firewall to remove unauthorized network access.

Here we demonstrate the traffic oscillator as an effective preventer of cyber-attacks. The first aspect is the re-organization of application transmission. Normal application design has very little restriction from the network, and tends to transmit information as soon as it is available. Thus, identifying the application packets and its relative payload size is impossible when there are unknown packet quantities being transmitted in the network. These packets are also multi-casted and arrive at random time intervals, making quantitative sampling inaccurate. Utilisation of the network oscillator assists the network in several ways. The number of individual packets is fewer and larger, making application packets more traceable. The resultant increased certainty means that routers and switches can reserve the switching space. The fixed time interval helps to establish a traffic monitoring refresh period. Retransmissions plan can be tailored based on network availability constraints rather than application convenience.

6.2.6 Results and discussion

This simulation represents an airport network topology carrying a diverse range of services. Although there is only one physical server (SWIM), and many clients in the network, the simulation splits three important applications (air-traffic control; radar information; additional services such as weather reports) into three virtual networks (Figure 6.5). This level of separation has not been customary in simulations to date, because application transmission cannot be separated easily without prior application knowledge. Here, we also include a hacker who has infiltrated the first virtual network

via one of the virtual application nodes, and who is attempting to disable the network by a DoS attack.

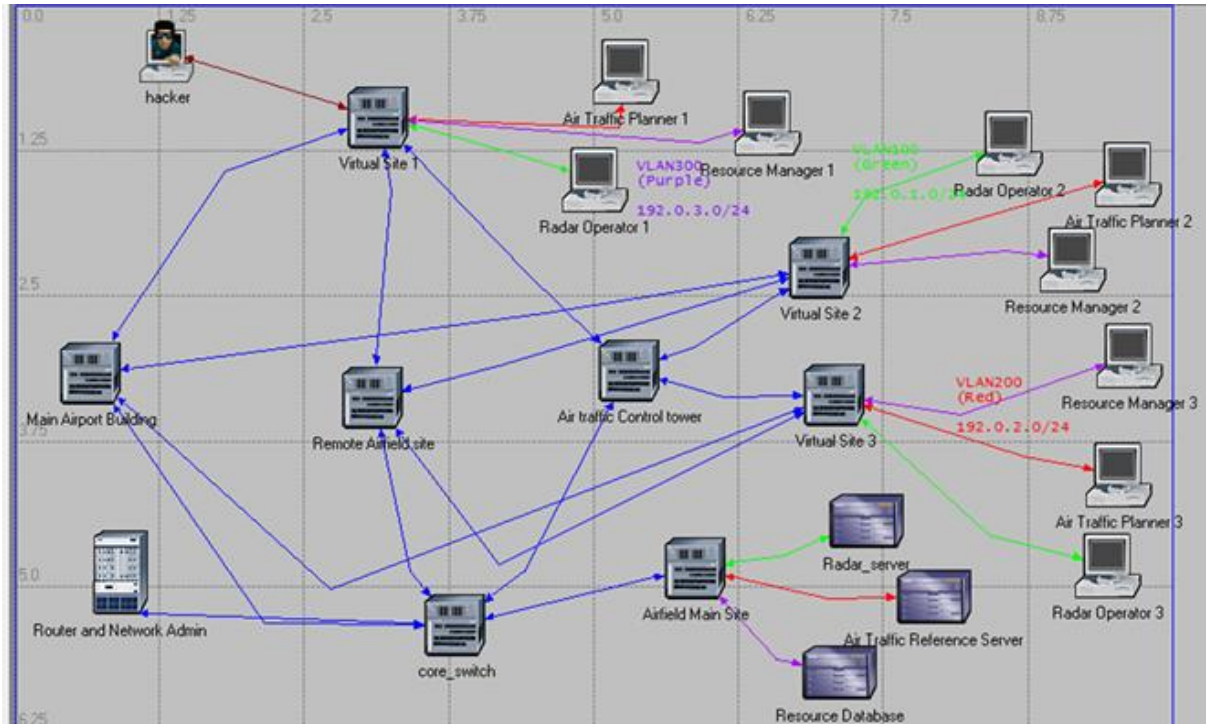


Figure 6.5 Each application is labelled as a separate end-device node with its virtual network to its respective server, a hacker has gained access to one of the virtual switches and begins to send a Denial of service attack, trying to bring down the whole network

6.2.7 Network Application Oscillators

The network oscillator not only controls application transmission in terms of time and payload, but will also build up prior-knowledge of each application transmission characteristic, which will help to identify and segregate causes and effects on the network. The airport network is improved by the provision of a constructive network switching plan for organising network traffic usage. This reduces the effectiveness of cyber-attacks by first imposing higher restrictions on new applications during normal

operation. The characteristics of such applications are identified, enabling the separation of cyber-attacks by reducing their transmission freedom. In this simulation, there are three types of communication: SWIM radar, radar operator, resource planner plus the hacker. The first of these increases its transmission based on the number of detectable aircraft and their properties [18] causing it to behaves like a series of short burst payloads. The other two, legitimate, applications regularly transfer updated air traffic flight plans, leading to periodic step function payload transmission. The hacker employs a DoS cyber-attack, designed to trigger network congestion and cause other services also to increase transmissions. The form of these transmissions resembles high intensity impulses. When each application transmission enters the airport network by means of its network oscillator, the FFT facilitates its display, as may be seen in Figure 6.7 and Figure 6.7.

6.2.8 Transmission Characteristics

Network traffic commonly conforms to one of two patterns, namely a burst of impulses or an on-going transfer of information (a step). Both these transmission characteristics are problematic in terms of traffic identification and differentiation. Burst transmission exhibits both random payload and random time intervals, whilst a step function has no particular pattern, especially when there are applications transmitting at the same payload per second. To illustrate that these transmissions become even more difficult when their characteristics change over time, we present the effect of regular periodic sampling in the standard network situation (Figure 6.6).

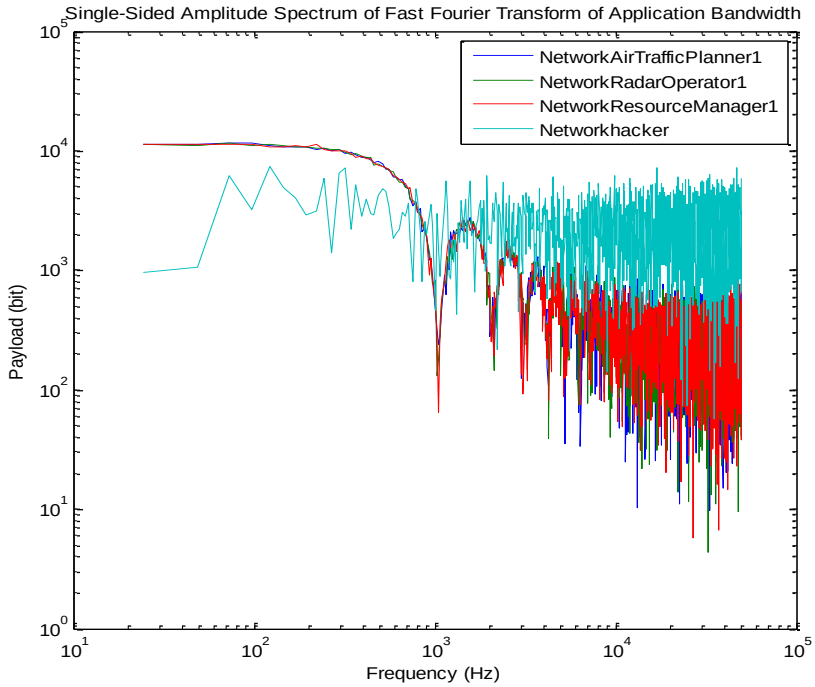


Figure 6.6 Discrete Fourier transform on the four applications communication, the hacker DoS attack shows up overloading all transmission payload even when the transmission is contained in the first virtual network

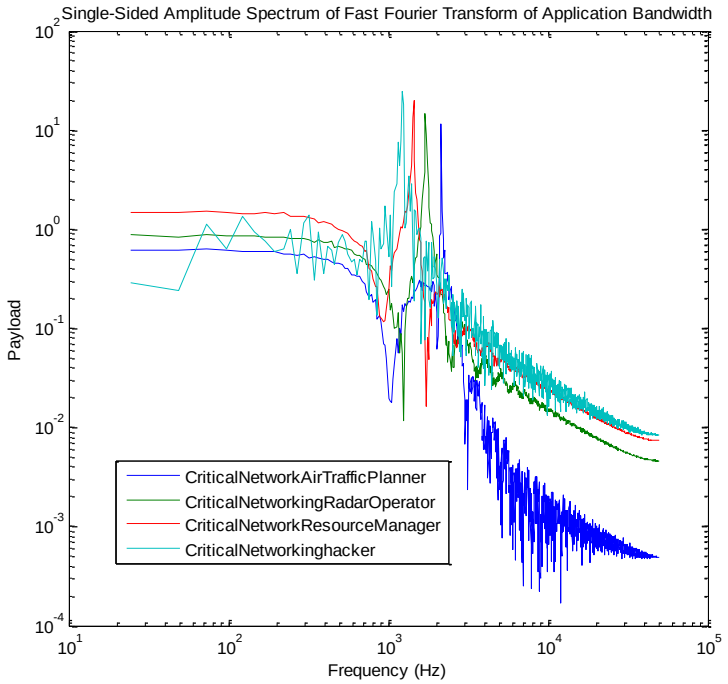


Figure 6.7 Discrete Fourier transform on the four applications communication, the hacker DoS attack shows up clearly using this analysis, this allows application firewall sampling, monitoring and filtering

As may be observed, this is ineffective in capturing detail, because each snapshot cannot model the rapidly changing traffic patterns. In contrast, the effect of first regulates a high intensity impulse transmission to a fixed payload step function by a large buffer, which is then fed into the network oscillator for upper and lower boundary payload transmission for easy capture by a traffic sampling tool. In Figure 6.6, the DoS cyber-attack is successful by building up network traffic congestion. The utilisation of the network oscillator means that the application transmissions in Figure 6.7 are unaffected by the cyber-attack, which can be easily removed by a selective band-pass payload filter. The previous transport protocol algorithm is only designed to operate for the benefits of its application connectivity and retransmission. To the hacker, retransmission presents a network weakness by triggering out-of-bound time window transmissions, which build up, causing a ramp-like transmission function of congestion, exponential delay time increase and buffer overflow. The DoS attack is much harder to separate from the normal transmission in Figure 6.7 because of its unknown high intensity multiple impulse transmission characteristic. This triggers a reaction as just described, resulting in network congestion, which is further amplified by other application retransmissions. Targeting cyber-attacks from the original transmission traffic is nearly impossible, but is made straightforward by the network oscillator.

6.3 Conclusions

We have presented a method of critical networking by taking a new view of the process of packet transmission. Germane to our approach it the realisation of the separation

between physical bandwidth and application bandwidth, coupled with the consideration of the application payload per packet. This insight permits the design of a combination of buffer, converter and flow controller that permits application layer filtering and control for the first time. This NTO allows a higher effective communication rate from servers to clients. By taking a step further than standard buffering by introducing the payload per packet ratio and the flow controller, the resonant NTO delivers a much less random packet stream. By delivering a fixed payload per packet and a more predicable response time, the method is ideal for real time, and safety critical communication. The simulation results include Ethernet for medium access control so offer the prospect of near-deterministic Ethernet. This is illustrated here by application to air traffic control radar data but the method is generic offering any control and monitoring task the possibility of employing standard Ethernet hardware but delivering safety critical performance.

In the past, real time monitoring has been ineffective in network monitoring and organization, because each application only operated for its own benefit, particularly when application transmissions were totally unmanaged and unrelated. The NTO device described here offers tailored operation parameters to every application, so that each has very distinctive payload sizes and transmission patterns. We have demonstrated the first framework for the classification of network traffic by packet, payload per packet and payload per second within sensitive networks using the NTO as a hybrid firewall. Patterns are observed in the Fourier domain, using the FFT illuminating application performance via fixed unified sampling intervals, thus revealing them without observing their overheads. This process facilitates identification of applications by their transmission characteristics (time-interval frequency) in addition to be header cross-correlation (self-identified packet system). Trojans, spyware and

any unauthorized application transmissions can no longer disguise themselves by fake protocol headers, because their transmission parameters are mismatched, and by not having been critically assigned.

The non-deterministic nature of the traffic is accommodated by the NTO by use of its buffer and flow controller in tandem to generate deterministic Ethernet patterns. Ultimately, this work addresses the fundamentals of critical networking research. the main goal of which is to reduce the overhead used in Ethernet traffic. This framework allows better understanding of each component. such as the difference between payload and packets.

To enable the discrimination between applications, the NTO acts as a ZOH, and a digital filter is placed at network checkpoints to identify small variances in application transmissions. A critical network device can additionally adjust its time-interval transmission by increasing/decreasing buffer sizes and flow rate. The effect of an increase in buffer size is to increase the hold-time and the relative payload per packet during the critical transmission. An increase in flow rate increases the hold time between critical transmissions, and spontaneously increases the level of payload per packet rate of change.

We have shown the utility of the approach by network simulation, which clearly illustrated that a cyber-attacker's traffic that would normally be buried, was revealed by use of the NTO. This new insight into network monitoring and security provides a toolkit for designing more complex networks, as opposed to more complex analysis tools.

Chapter 7 – Conclusions and Further Work

Peer-to-peer communication has no congestion delay; payload packet exchange without a network will only experience propagation delay. Additional delay is caused by buffering where the available bandwidth cannot be supported by the application demand. The introduction of networking allows more nodes to be connected to each other using the same physical medium, and at the same time network congestion delay has also arisen from packets waiting to be served. There will always be congestion delay when the service rate requires higher physical bandwidth than the network can provide. Congestion delay can only be eliminated when the network workload (payload per second) is managed and operating within the physical bandwidth. Network management can be viewed from two perspectives, a maintenance based network, and an application based network. The former has no latency but is highly inflexible for swapping different applications. Applications running over maintenance based networks are designed for continuous throughput service (consecutive transmission) but only support a limited number of application streams. The latter, became the de-facto framework for modern “Ethernet” networking as a result of its simple segregation of network tasks (OSI) and flexibility for new network technology such as new physical bandwidth and network protocols, for example swapping between optical cables and radio waves. Network management has consequentially become a best effect delivery system, where the first attempt at communication is not always guaranteed. The problem of packets lost in the network is no longer a problem with high bit error rates of physical links, but rather the mismanagement of transmission and time windowing. Packets can be miss-directed by routers, be lost in the queue though trip time out, and lost by dropout (exceeding

the receiver response time). These three results can be created from infinitely many combinations of unknown network conditions, especially when the network has multiple queues, paths and service rates. In this work, a deterministic approach called Critical networking has been developed. The main design for Critical Networking is to isolate the desire for applications to use network resource in a free-for-all fashion, and thus eliminate computation competition where applications are encouraged to send redundant packets in an optimal way to improve connectivity.

The overall objective originally was to rationalise and compress the knowledge of networking with a simple model for packet transmission. The main barrier for any congestion control scheme is the unknown initial state problem, which is related to chaos theory. Because of unknown initial state, many congestion problems are randomly built up, even when all the protocol and transmission management are man-made, designed and follow an algorithm. The design of an intelligent packet monitoring system based on support vector machines and geometric time windowing with Lagrange multipliers is to compress packet data into traffic model. This sheds light onto a packet transmission congestion model which behaves as arithmetic expansion when the initial packet was blocked in a network. This expansion can be cured by the application of additional serving bandwidth to remove the build-up of packets, thus creating an alternate view for active traffic theory management as shown in chapter 4. When the initial packet transmission was also fixed in a deterministic state (Laplacian system model), the NTO has been created to demonstrate the power of congestionless and bufferless networking. The NTO combines discrete packets into a ZOH traffic management scheme for interesting applications such as security (firewalling) and maintenance (long distance zero congestion network).

Moreover, a more effective way of communicating is to remove the need for a transport window (communication feedback model), from which errors are expected and which is likely to be wrong during the iteration process of finding the correct time window. Critical networking thus removes the likely repeated iterations resulting from changes in network resources - these errors in communication reduce the physical bandwidth serving capacity of the network. Multi-casting also reduces service rate gain from a multipath network (an increase in supplement of bandwidth from other links). The solution for creating deterministic Ethernet network is to introduce a network traffic oscillator (NTO) for each application. Each application has a peak (requires maximum serving throughput to catch up with the requirement of application traffic -payload per second), and a trough that provides a gap for other network applications to occupy. The NTO uses a digital stepped sine wave, the shape of which depends on the ratio of dividing the bandwidth by the frequency division block. This thus facilitates the management of the network in the maintenance style, and therefore no latency (congestion delay) is experience per application, whilst keeping the philosophy of flexibility amongst other multimedia applications.

In retrospect, when a network service is tolerated by applications and users, network congestion problems are hidden, particularly by increasing the physical link technology so that small congestion delays are not noticeable; the problem is not thereby resolved, but rather its impact on the applications and users has been reduced. The cure of any congestion delay problem is made more difficult when it is also random, and congestion problems are very different on a per network basis.

Historically, quick network solutions to these rapid congestion problems have been developed such as another protocol or expanding another area of study to tackle the task, but these offer diminishing returns when there is no extra bandwidths to

support this service. The cause of congestion delay is the unaccountable extra network payload itself and when there is deterministic payload transmission, the network has zero congestion. This problem is further escalated when each payload increases in size due to standards and regulation (packet overheads) and more payloads are generated for application packet redundancy. When each application operates within its service requirement boundary (payload per second), there is no network congestion. Random packet transmission leads to random packet queuing and serving inside a node, these extra queuing and serving requirements can be removed by eliminating any uncertainty of random network condition from node to node. Erlang traffic theory is useful for mapping the probability of service, buffering and congestion delay for packet transmission when the statistical arriving and service rate is known. However, it does not support the multi-queue management because the fact that serving and arrival rate are random, it is as effective as one shorter queue; in the eye of a network traffic engineer, it is better to have one long queue. However, when the service and arrival rates are known, each service rate can selectively be managed and catered for before the extra congestion load has arrived, making congestion delay disappear using a Laplacian of modelling past and future arrival and service rate. This study has been carried out extensively in chapter four, where an arrival and service model was used to predict the serving and waiting probability; moreover it was also demonstrated that superior performance results from actively managing the arrival rate, the queue and the serving rate.

The NTO in this research has two variations, a system model based on Laplacian differentiation changes of payload inside a queue, and a discrete packet/payload model for discrete event based simulation. The latter has been implemented to be evaluated inside the OPNET network simulator and ultimately in a physical world. The

benefit of the NTO is to remove extra optimisation between link capacity and payload transmission required to compensate for any additional congestion experience in a network. The NTO also separates each network application service so there are no collisions between packet transmissions.

Further work may include implementing the use of NTO in a mobile platform where there is a higher level of uncertainty such as distances, bandwidth and bit error rate. Moreover, the necessary information for critical networking may not be readily available. However, higher encryption and forward error code correction can aid the delivery of information correctly in the first instance, and remove any unnecessary network, transport control protocol that consumes the limited bandwidth. The limitation of deterministic Ethernet network is the limit on the knowledge in network management.

References

[1] Y. Langeron, A. Barros, A. Grall, and C. Berenguer, "Combination of safety integrity levels (SILs): A study of IEC61508 merging rules," *Journal of Loss Prevention in the Process Industries,* vol. 21, pp. 437-449, 2008.

[2] R. Bell, "Design essentials to achieve a specified SIL," in *5th IET Seminar on SIL Determination*, 2009, pp. 1-26.

[3] A. Sipe and J. Moore, "Air traffic functions in the NextGen and SESAR airspace," in *IEEE/AIAA 28th Digital Avionics Systems Conference*, 2009, pp. 2.A.6-1-2.A.6-7.

[4] B. Haindl, C. Rihacek, M. Sajatovic, B. Phillips, J. Budinger, M. Schnell*, et al.*, "Improvement of L-DACS1 design by combining B-AMC with P34 and WiMAX technologies," in *Integrated Communications, Navigation and Surveillance Conference*, 2009, pp. 1-8.

[5] B. Haindl, C. Rihacek, and M. Sajatovic, "Integrated navigation and surveillance capability within L-DACS1," in *IEEE/AIAA 28th Digital Avionics Systems Conference*, 2009, pp. 4.B.6-1-4.B.6-11.

[6] R. Graham, N. Pilon, L. Tabernier, H. Koelman, and P. Ravenhill, "Performance framework and influence model in ATM," in *IEEE/AIAA 28th Digital Avionics Systems Conference*, 2009, pp. 2.A.5-1-2.A.5-11.

[7] J. D. Day and H. Zimmermann, "The OSI reference model," *Proceedings of the IEEE,* vol. 71, pp. 1334-1340, 1983.

[8] M. Rees, "The EUROCONTROL Surveillance Strategy," in *Tyrrhenian International Workshop on Digital Communications - Enhanced Surveillance of Aircraft and Vehicles*, 2008, pp. 1-5.

[9] M. Cano, P. Sanchez-Escalonilla, and M. M. Dorado, "Complexity analysis in the next generation of air traffic management system," in *IEEE/AIAA 26th Digital Avionics Systems Conference*, 2007, pp. 3.D.4-1-3.D.4-9.

[10] L. Jian, Z. Long, and Y. JianGuo, "AFDX based avionic data bus architecture design and analysis," in *International Symposium on Autonomous Decentralized Systems*, 2009, pp. 1-1.

[11] D. Eier and W. Kampichler, "Eurocae WG-67 standards for voice-over-IP in ATM for advanced NEXTGEN conops," in *Integrated Communications Navigation and Surveillance Conference (ICNS),* , 2010, pp. C8-1-C8-9.

[12] T. N. Saadawi and A. H. Abdelmonem, "Ed-net; A multi-Protocol educational local area network," *IEEE Communications Magazine,* vol. 25, pp. 34-40, 1987.

[13] R. Bannatyne, "Time triggered protocol-fault tolerant serial communications for real-time embedded systems," in *Wescon*, 1998, pp. 86-91.

[14] M. Kunes and T. Sauter, "Fieldbus-internet connectivity: the SNMP approach," *IEEE Transactions on Industrial Electronics,* vol. 48, pp. 1248-1256, 2001.

[15] B. Black, "SIL determination requirements and problem areas," in *The IEE Seminar on Methods and Tools for SIL Determinationa*, 2005, pp. 1-1/15.

[16] J. Lopez, M. Vilaplana, I. Bayraktutar, J. Klooster, J. M. Asensio, G. McDonald*, et al.*, "Towards an open test bed for the study of trajectory synchronization in the future ATM system: The ASIS initiative," in *Integrated Communications, Navigation and Surveillance Conference*, 2009, pp. 1-14.

[17] A. Mathias and M. Hess, "Machine-readable encoding standard specifications in ATC," in *Digital Communications - Enhanced Surveillance of Aircraft and Vehicles (TIWDC/ESAV), Tyrrhenian International Workshop*, 2011, pp. 241-246.

[18] R. Houdebert and B. Ayral, "Making SWIM interoperable between US and Europe," in *Integrated Communications Navigation and Surveillance Conference (ICNS)*, 2010, pp. C4-1-C4-8.

[19] D. D. Crescenzo, A. Strano, and G. Trausmuth, "SWIM: A Next Generation ATM Information Bus-The SWIM-SUIT Prototype," in *Enterprise Distributed Object Computing Conference Workshops, 14th IEEE International*, 2010, pp. 41-46.

[20] F. Schreckenbach, K. Leconte, C. Baudoin, C. Kissling, C. Bauer, and S. Ayaz, "Functional building blocks for an integrated aeronautical IP-network," in *Integrated Communications, Navigation and Surveillance Conference,* , 2008, pp. 1-9.
[21] D. Boggs, J. F. Shoch, E. Taft, and R. Metcalfe, "Pup: An Internetwork Architecture," *IEEE Transactions on Communications,* vol. 28, pp. 612-624, 1980.
[22] J. D'Ambrosia, "100 gigabit Ethernet and beyond [Commentary]," *IEEE Communications Magazine,* vol. 48, pp. S6-S13, 2010.
[23] J. F. K. a. K. W. Ross, *Computer Networking*, Second Edition ed.: Addison Wesley 2003.
[24] H. Zhigang, Y. Yansong, and S. Ninghui, "High performance Sockets over kernel level virtual interface architecture," in *Proceedings. Eighth International Conference on High-Performance Computing in Asia-Pacific Region*, 2005, p. pp. 226.
[25] H.-W. Jin and C. Yoo, "Impact of protocol overheads on network throughput over high-speed interconnects: measurement, analysis, and improvement," *Journal of Supercomputing,* vol. 41, pp. 17-40, 2007.
[26] A. L. a. K. Altis, "World-Wide Web proxies," *Computer Networks and ISDN Systems,* vol. vol. 27, no. 2, pp. pp. 147–154, 1994.
[27] M. A. Arfeen, K. Pawlikowski, and A. Willig, "A Framework for Resource Allocation Strategies in Cloud Computing Environment," in *IEEE 35th Annual Computer Software and Applications Conference Workshops* 2011, pp. 261-266.
[28] G. Ciaccio, "Messaging on gigabit ethernet: some experiments with GAMMA and other systems," in *Proceedings 15th International Parallel and Distributed Processing Symposium.*, 2001, pp. 1624-1631.
[29] P. B. H.-W. Jin, C. Yoo, J.-Y. Choi, and D. K. Panda, "Exploiting NIC architectural support for enhancing IP based protocols on high performance networks," *Journal of Parallel and Distributed Computing,* vol. vol. 65, no. 11, pp. pp. 1348–1365, 2005.
[30] C. Song, L. San-qi, and J. Ghosh, "Predictive dynamic bandwidth allocation for efficient transport of real-time VBR video over ATM," *IEEE Journal on Selected Areas in Communications,* vol. 13, pp. 12-23, 1995.
[31] T. Arai and H. Habuchi, "A comparison of the M-ary/SSMA ALOHA system and the DS/SSMA ALOHA system," in *The 11th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2000, pp. 1530-1534 vol.2.
[32] Z. Tsai, W. d. Wang, and J. F. Chang, "A dynamic bandwidth allocation scheme for ATM networks," in *Twelfth Annual International Phoenix Conference on Computers and Communications*, 1993, pp. 289-295.
[33] M. Chiani, E. Milani, and R. Verdone, "A semi-analytical approach for performance evaluation of TCP-IP based mobile radio links," in *IEEE Global Telecommunications Conference*, 2000, pp. 937-942 vol.2.
[34] R. Obermaisser, "Reuse of CAN-Based Legacy Applications in Time-Triggered Architectures," *IEEE Transactions on Industrial Informatics,* vol. 2, pp. 255-268, 2006.
[35] A. Rindos, S. Woolet, L. Nicholson, and M. Vouk, "A performance evaluation of emerging Ethernet technologies: switched/high-speed/full-duplex Ethernet and Ethernet LAN emulation over ATM," in *Proceedings of IEEE Southeastcon Bringing Together Education, Science and Technology*, 1996, pp. 401-404.
[36] A. Mifdaoui, F. Frances, and C. Fraboul, "Performance Analysis of a Master/Slave Switched Ethernet for Military Embedded Applications," *IEEE Transactions on Industrial Informatics,* vol. 6, pp. 534-547, 2010.
[37] A. Mifdaoui, F. Frances, and C. Fraboul, "Real-time characteristics of Switched Ethernet for "1553B"-Embedded Applications: Simulation and Analysis," in *SIES. International Symposium on Industrial Embedded Systems*, 2007, pp. 33-40.

[38] H. Li, H. Zhang, F. Xia, and W. Huang, "Research on Communication Controller between FlexRay and Modbus," in *International Conference on Signal Acquisition and Processing*, 2009, pp. 73-76.
[39] R. D. Love, "Specifying the physical layer in a LAN standard: a comparison of CSMA/CD and token ring," in *Proceedings of the 13th Conference on Local Computer Networks*, 1988, pp. 146-150.
[40] P. Parag and J. F. Chamberland, "Queueing Analysis of a Butterfly Network for Comparing Network Coding to Classical Routing," *IEEE Transactions on Information Theory,* vol. 56, pp. 1890-1908, 2010.
[41] P. Sharma and M. Devgan, "Virtual Device Context &;Securing with Scalability and Cost Reduction," *IEEE Potentials,* vol. 31, pp. 35-37, 2012.
[42] J. Jasperneite, P. Neumann, M. Theis, and K. Watson, "Deterministic real-time communication with switched Ethernet," in *Factory Communication Systems, 4th IEEE International Workshop on*, 2002, pp. 11-18.
[43] M. Veeraraghavan and M. Karol, "Internetworking connectionless and connection oriented networks," *Communications Magazine, IEEE,* vol. 37, pp. 130-138, 1999.
[44] M. Ehammer, T. Graeupl, C. H. Rokitansky, and C. Kissling, "The operation of TCP over aeronautical networks," in *Integrated Communications, Navigation and Surveillance Conference*, 2009, pp. 1-11.
[45] R. Schmidt, E. Rawson, R. Norton, Jr., S. Jackson, and M. Bailey, "Fibernet II: A Fiber Optic Ethernet," *Selected Areas in Communications, IEEE Journal on,* vol. 1, pp. 702-711, 1983.
[46] E. Nikoloutsos, A. Prayati, A. P. Kalogeras, V. Kapsalis, S. Koubias, and G. Papadopoulos, "Integrating IP traffic into fieldbus networks," in *Proceedings of the IEEE International Symposium on Industrial Electronics* 2002, pp. 67-72 vol.1.
[47] M. Xiao, "MODBUS Protocol Converter Which Used in 220KV Main Transformer's Air-Cooled Control," in *2nd International Workshop on Database Technology and Applications* 2010, pp. 1-4.
[48] "Conformance Test Methodology for IEEE Standards for Local and Metropolitan Area Networks: Supplement to Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications: Type 10BaseT MAU Conformance Test Methodology (Section 6) (Includes related changes to ISO/IEC 8802-3 : [ANSI/IEEE Std 802.3, Edition])," *IEEE Std 1802.3d-1993,* p. 1, 1994.
[49] "IEEE Standards for Local Area Networks: Supplements to Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications," *ANSI/IEEE Std 802.3a,b,c, and e,* p. 0_1, 1987.
[50] G. Brunello, R. Smith, and C. B. Campbell, "An application of a protective relaying scheme over an ethernet LAN/WAN," in *Transmission and Distribution Conference and Exposition, IEEE/PES*, 2001, pp. 522-526
[51] V. Vermeer, "Wireless LANs; why IEEE 802.11 DSSS?," in *Conference Proceedings Wescon* 1997, pp. 172-178.
[52] J. C. Becker, B. Nitzberg, and R. F. Van der Wijngaart, "Predicting cost/performance trade-offs for Whitney: a commodity computing cluster," in *Proceedings of the Thirty-First Hawaii International Conference on System Sciences*, 1998, pp. 504-513.
[53] H. Miyata and M. Endo, "Design and evaluation of a fieldbus protocol over IPv6," in *Industrial Informatics (INDIN), IEEE Conference*, 2010, pp. 136-141.
[54] T. Wei-ming and G. Hong-wei, "A Fieldbus Comprehensive Evaluation and Selection Method Based on Entropy-Ideal Point," in *Intelligent Systems Design and Applications. Eighth International Conference*, 2008, pp. 564-568.
[55] W. Haifang, R. Yu, L. Shengtao, and C. Jinhua, "Fieldbus technology and rolling process automation," in *Computer Design and Applications (ICCDA)*, 2010, pp. V4-73-V4-76.

[56] W. Zhongfeng, Y. Haibin, W. Hong, X. Aidong, and Z. Yue, "The research of XML-based extensible device description for FF fieldbus system," in *30th Annual Conference of IEEE Industrial Electronics Society*, 2004, pp. 2600-2603 Vol. 3.

[57] P. Zin-Won and K. Myung-Kyun, "Design of an integrated fieldbus gateway," in *The 9th Russian-Korean International Symposium on Science and Technology*, 2005, pp. 843-846.

[58] S. Hu, Z. Zhao, Y. Zhang, and S. Wang, "A novel Modbus RTU-based communication system for adjustable speed drives," in *IEEE Vehicle Power and Propulsion Conference*, 2008, pp. 1-5.

[59] C. Bo and X. Guangbin, "Design and realization of an intelligent data acquisition and display system based on AT89C52 and modbus," in *International Colloquium on Computing, Communication, Control and Management*, 2009, pp. 455-458.

[60] L. Chen, "Research on Voltage Acquisition System Based on Modbus Industrial Bus and Single Chip," in *International Conference on Signal Processing Systems*, 2009, pp. 585-588.

[61] L. Gen-Yih, C. Yu-Jen, L. Wen-Chung, and C. Tsung-Chieh, "Toward Authenticating the Master in the Modbus Protocol," *IEEE Transactions on Power Delivery,* vol. 23, pp. 2628-2629, 2008.

[62] J. Dudak and P. Cicak, "CPN model of the MODBUS protocol," in *MECHATRONIKA, 13th International Symposium*, 2010, pp. 43-45.

[63] P. Dao-gang, Z. Hao, Y. Li, and L. Hui, "Design and Realization of Modbus Protocol Based on Embedded Linux System," in *International Conference on Embedded Software and Systems Symposia*, 2008, pp. 275-280.

[64] H. Shengnan, Y. Zheng, and W. Zhaohua, "Optimize and design fieldbus network based embedded systems," in *2nd International Conference on Industrial and Information Systems*, 2010, pp. 420-423.

[65] X. Tu, "Design of Air Compressor Monitoring System Based on Modbus Protocol," in *Electrical and Control Engineering*, 2010, pp. 710-713.

[66] Y. Liu, C. Wang, C. Yu, and X. Qiao, "Research on ZigBee Wireless Sensors Network Based on ModBus Protocol," in *International Forum on Information Technology and Applications*, 2009, pp. 487-490.

[67] G. Lou, H. Zhang, and W. Zhao, "Research on designing method of CAN bus and Modbus protocol conversion interface," in *International Conference on Future BioMedical Information Engineering*, 2009, pp. 180-182.

[68] M. J. Solvie, "Configuration of distributed time-critical fieldbus systems," in *Proceedings of 2nd International Workshop on Configurable Distributed Systems*, 1994, p. 211.

[69] G. Shoshani, S. Mitschke, and S. Stephan, "Industrial Fieldbus technology and Fieldbus cable overview 2014; Cable standards and electrical qualifications," in *Petroleum and Chemical Industry Conference (PCIC), Record of Conference Papers Industry Applications Society 57th Annual*, 2010, pp. 1-10.

[70] X.-l. Zhang, "Study on communication scheduling of fieldbus," in *Control and Decision Conference*, 2009, pp. 565-570.

[71] R. P. Pantoni, N. M. Torrisi, D. Brandao, and E. A. Mossin, "Integration of an open and non-proprietary device description technology in a foundation fieldbus simulator," in *IEEE International Workshop Factory Communication Systems*, 2008, pp. 435-444.

[72] H. Bauer, J. L. Scharbarg, and C. Fraboul, "Applying and optimizing trajectory approach for performance evaluation of AFDX avionics network," in *IEEE Conference on Emerging Technologies & Factory Automation*, 2009, pp. 1-8.

[73] C. Xin, X. Xudong, and W. Jianxiong, "A Software Implemetation of AFDX End System," in *NISS International Conference on New Trends in Information and Service Science*, 2009, pp. 558-563.

[74] I. Khazali, M. Boulais, and P. Cole, "AFDX software network stack implementation & Practical lessons learned," in *Digital Avionics Systems Conference, DASC . IEEE/AIAA 28th*, 2009, pp. 1.B.5-1-1.B.5-10.

[75] J. L. Scharbarg, F. Ridouard, and C. Fraboul, "A Probabilistic Analysis of End-To-End Delays on an AFDX Avionic Network," *IEEE Transactions on Industrial Informatics,* vol. 5, pp. 38-49, 2009.
[76] H. Bauer, J. L. Scharbarg, and C. Fraboul, "Improving the Worst-Case Delay Analysis of an AFDX Network Using an Optimized Trajectory Approach," *IEEE Transactions on Industrial Informatics,* vol. 6, pp. 521-533, 2010.
[77] L. Xiaoting, J. L. Scharbarg, and C. Fraboul, "Improving end-to-end delay upper bounds on an AFDX network by integrating offsets in worst-case analysis," in *IEEE Conference on Emerging Technologies and Factory Automation*, 2010, pp. 1-8.
[78] Z. Xingxing and S. Dong, "The Research on End-to-End Delay Calculation Method for Real-Time Network AFDX," in *CiSE International Conference on Computational Intelligence and Software Engineering*, 2009, pp. 1-4.
[79] J. Zhang, D. Li, and Y. Wu, "Modelling and performance analysis of AFDX based on Petri Net," in *2nd International Conference on Future Computer and Communication* 2010, pp. V2-566-V2-570.
[80] D.-j. Li, J.-d. Zhang, and B. Liu, "Periodic message-based modeling and performance analysis of AFDX," in *IEEE International Conference on Wireless Communications, Networking and Information Security*, 2010, pp. 162-166.
[81] D. Song, X. Zeng, L. Ding, and Q. Hu, "The Design and Implementation of the AFDX Network Simulation System," in *International Conference on Multimedia Technology*, 2010, pp. 1-4.
[82] L. Ding, D. Song, X. Zeng, and Q. Hu, "The Research of AFDX System Simulation Model," in *International Conference on Multimedia Technology*, 2010, pp. 1-4.
[83] M. Anand, S. Vestal, S. Dajani-Brown, and L. Insup, "Formal modeling and analysis of the AFDX frame management design," in *Ninth IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing*, 2006, p. 7 pp.
[84] F. Brajou and P. Ricco, "AFDX-based flight test computer concept," *IEEE Instrumentation & Measurement Magazine,* vol. 8, pp. 55-58, 2005.
[85] F. Brajou and P. Ricco, "The Airbus A380 - an AFDX-based flight test computer concept," in *Proceedings AUTOTESTCON* 2004, pp. 460-463.
[86] EUROCONTROL, "Eurocontrol Guidelines for implementation support (EGIS) Part 5 Communication and Navigation Specifications Chapter 12 Surveillance " EUROCONTROL, Report15/03/07 2007.
[87] D. De Smedt and T. Putz, "Flight simulations using time control with different levels of flight guidance," in *IEEE/AIAA 28th Digital Avionics Systems Conference*, 2009, pp. 2.C.5-1-2.C.5-15.
[88] M. R. C. Jackson, J. Gonda, R. Mead, and G. Saccone, "The 4D trajectory data link (4DTRAD) service - Closing the loop for air traffic control," in *Integrated Communications, Navigation and Surveillance Conference*, 2009, pp. 1-10.
[89] D. Medina, F. Hoffmann, S. Ayaz, and C. H. Rokitansky, "Topology characterization of high density airspace aeronautical ad hoc networks," in *5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, 2008, pp. 295-304.
[90] J. F. Porras and M. Parra, "Atm initiatives on reduced separation minima," in *IEEE/AIAA 26th Digital Avionics Systems Conference*, 2007, pp. 3.C.4-1-3.C.4-12.
[91] G. Gawinowski, F. Drogoul, R. Guerreau, R. Weber, and J. L. Garcia, "ERASMUS contribution to the 2020 SESAR scenario," in *DASC IEEE/AIAA 27th on Digital Avionics Systems Conference*, 2008, pp. 3.A.2-1-3.A.2-10.
[92] D. Di Crescenzo, A. Strano, and G. Trausmuth, "System Wide Information Management: The SWIM-SUIT prototype," in *Integrated Communications Navigation and Surveillance Conference* 2010, pp. C2-1-C2-13.
[93] D. Fowler, E. Perrin, and R. Pierce, "Safety assessment of the SESAR Operational Concept - European air traffic management in 2020 and beyond," in *4th IET International Conference on Systems Safety . Incorporating the SaRS Annual Conference*, 2009, pp. 1-8.

[94] D. Fowler, P. Mana, and B. Tiemeyer, "Safety Management on a European Scale," in *The 1st Institution of Engineering and Technology International Conference on System Safety*, 2006, pp. 207-216.
[95] V. Srivastava, C. Wargo, and S. Lai, "Aviation application over IPv6: performance issues," in *IEEE Aerospace Conference* 2004, p. 1670 Vol.3.
[96] R. Kolle and A. Tarter, "Towards an Aviation Security Incident Management capability in NextGen and SESAR," in *Integrated Communications, Navigation and Surveillance Conference*, 2009, pp. 1-11.
[97] A. G. King, "SIL Determination and the application of LOPA," in *SIL Determination - Minimising the Risk of Your Systems, 2008 4th IET Seminar on*, 2008, pp. 1-1.
[98] D. Fowler, E. Perrin, and R. Pierce, "Success is not merely absence of failure - a systems-engineering approach to safety assessment," in *4th IET International Conference on Systems Safety Incorporating the SaRS Annual Conference,* , 2009, pp. 1-6.
[99] R. Bell, "Design essentials to achieve a specified SIL," in *SIL Determination - Minimising the Risk of Your Systems, 4th IET Seminar on*, 2008, pp. 1-49.
[100] J. Brazendale, "Regulatory issues in functional safety," in *IET Seminar on Safety Assessments: When is enough enough?*, 2010, pp. 1-29.
[101] R. Bell, "Design essentials to achieve a specified SIL," in *SIL Determination - Minimising the Risk of Your Systems, 2008 4th IET Seminar on*, 2008, pp. 1-49.
[102] J. E. Ball, J. Feldman, J. R. Low, R. Rashid, and P. Rovner, "RIG, Rochester's Intelligent Gateway: System Overview," *IEEE Transactions on Software Engineering,* vol. SE-2, pp. 321-328, 1976.
[103] X. Daojun, Q. Yang, and S. Chee Kheong, "Deterministic QoS Provisioning with Network Calculus Based Admission Control in WDM EPON Networks," in *IEEE International Conference on Communications*, 2009, pp. 1-6.
[104] C. F. Ahmed, S. K. Tanbeer, and B.-S. Jeong, "A Framework for Mining High Utility Web Access Sequences," *Iete Technical Review,* vol. 28, pp. 3-16, Jan-Feb 2011.
[105] Q. Liao, A. Blaich, D. VanBruggen, and A. Striegel, "Managing networks through context: Graph visualization and exploration," *Computer Networks,* vol. 54, pp. 2809-2824, Nov 15 2010.
[106] C. F. Ahmed, S. K. Tanbeer, and B.-S. Jeong, "A Novel Approach for Mining High-Utility Sequential Patterns in Sequence Databases," *Etri Journal,* vol. 32, pp. 676-686, Oct 2010.
[107] G. C. Lan, T. P. Hong, and V. S. Tseng, "Reducing Database Scans for On-shelf Utility Mining," *Iete Technical Review,* vol. 28, pp. 103-112, Mar-Apr 2011.
[108] D. Starobinski, M. Karpovsky, and L. A. Zakrevski, "Application of network calculus to general topologies using turn-prohibition," *Networking, IEEE/ACM Transactions on,* vol. 11, pp. 411-421, 2003.
[109] T. K. Sarkar, H. Schwarzlander, C. Seungwon, M. S. Palma, and M. C. Wicks, "Stochastic versus deterministic models in the analysis of communication systems," *IEEE Antennas and Propagation Magazine,* vol. 44, pp. 40-50, 2002.
[110] R. K. Mallik, M. R. Bhatnagar, and J. H. Winters, "Deterministic processing for MIMO systems," in *4th International Symposium on Communications, Control and Signal Processing*, 2010, pp. 1-5.
[111] L. E. Smith, C. J. Gesh, R. T. Pagh, R. J. McConn, J. E. Ellis, W. R. Kaye*, et al.*, "Deterministic Transport Methods for the Simulation of Gamma-Ray Spectroscopy Scenarios," in *IEEE Nuclear Science Symposium Conference Record*, 2006, pp. 588-592.
[112] E. Gascard, "From Sequential Extended Regular Expressions to Determinstic Finite Automata," in *Information and Communications Technology, 2005. Enabling Technologies for the New Knowledge Society: ITI 3rd International Conference on*, 2005, pp. 145-157.

[113] Z. Yunzhou, X. Dingyu, W. Chengdong, J. Peng, and C. Long, "Research of nodes deployment for wireless sensor network in determinstic area," in *2nd IEEE International Conference on Computer Science and Information Technology*, 2009, pp. 149-153.
[114] S. S. R. Abidi and J. Ong, "A data mining strategy for inductive data clustering: a synergy between self-organising neural networks and K-means clustering techniques," in *TENCON Proceedings* 2000, pp. 568-573 vol.2.
[115] K. M. Rogers and T. J. Overbye, "Clustering of Power System Data and Its Use in Load Pocket Identification," in *44th Hawaii International Conference on System Sciences (HICSS)*, 2011, pp. 1-10.
[116] K. Tasdemir, Mer, x000E, and E. nyi, "A Validity Index for Prototype-Based Clustering of Data Sets With Complex Cluster Structures," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. PP, pp. 1-15, 2011.
[117] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda*, et al.*, "Top 10 algorithms in data mining," *Knowledge and Information Systems,* vol. 14, pp. 1-37, Jan 2008.
[118] J. F. I. Elder, "Top 10 data mining mistakes," in *Fifth IEEE International Conference on Data Mining,*, 2005, p. 1 pp.
[119] S. Wei, "A Novel Hybrid GA Based SVM Short Term Load Forecasting Model," in *Second International Symposium on Knowledge Acquisition and Modeling*, 2009, pp. 227-229.
[120] M. Ponjavic and A. Karabegovic, "Pareto-based genetic algorithm in multi-objective geospatial analysis," in *Proceedings of the 33rd International Convention MIPRO*, 2010, pp. 680-685.
[121] I. Papaefstathiou, "Titan II: an IPComp processor for 10Gbit/sec networks," in *IEEE Computer Society Annual Symposium on VLSI*, 2003, pp. 234-235.
[122] F. Shurui, S. Hexu, and L. Jie, "On application method of Network Calculus to upper delay bound research," in *3rd IEEE International Conference on Computer Science and Information Technology*, 2010, pp. 566-570.
[123] K. L. Kaiser, *Transmission Lines, Matching, and Crosstalk*: CRC Press, 2005.
[124] M. Molle, M. Kelkunte, and J. Kadambi, "Frame bursting: a technique for scaling CSMA/CD to gigabit speeds," *Network, IEEE,* vol. 11, pp. 6-15, 1997.
[125] M. Nesenbergs, "A Hybrid of Erlang B and C Formulas and Its Applications," *Communications, IEEE Transactions on,* vol. 27, pp. 59-68, 1979.
[126] J. Farserotu and A. Tu, "TCP/IP over low rate ATM-SATCOM links," in *IEEE Military Communications Conference,* , 1996, pp. 162-167 vol.1.
[127] M. A. Arfeen, K. Pawlikowski, and A. Willig, "A Framework for Resource Allocation Strategies in Cloud Computing Environment," in *IEEE 35th Annual Computer Software and Applications Conference Workshops (COMPSACW),*, 2011, pp. 261-266.
[128] C. Yuh-Shyan, H. Chih-Shun, and Y. Wei-Han, "An IP Passing Protocol for Vehicular Ad Hoc Networks with Network Fragmentations," in *Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing* 2011, pp. 49-56.
[129] A. Belenky and N. Ansari, "Accommodating fragmentation in deterministic packet marking for IP traceback," in *IEEE Global Telecommunications Conference* 2003, pp. 1374-1378 vol.3.
[130] J. Chu, "Deterministic and Real-Time Communication over Ethernet in EPA Control System," in *The Sixth World Congress on Intelligent Control and Automation*, 2006, pp. XI-XII.
[131] S. Rajasegarar, A. Shilton, C. Leckie, R. Kotagiri, and M. Palaniswami, "Distributed training of multiclass conic-segmentation support vector machines on communication constrained networks," in *Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 2010, pp. 211-216.
[132] L. Bruzzone, M. Marconcini, and C. Persello, "Fusion of spectral and spatial information by a novel SVM classification technique," in *IEEE International Geoscience and Remote Sensing Symposium*, 2007, pp. 4838-4841.

[133] G. Carvajal, W. Chun Wah, and S. Fischmeister, "Evaluation of Communication Architectures for Switched Real-Time Ethernet," *IEEE Transactions on Computers,* vol. 63, pp. 218-229, 2014.
[134] Z. Zhen-Dong, L. Yun-Yong, N. Jun-Hong, and Z. Jing, "RBF-SVM and its application on reliability evaluation of electric power system communication network," in *Machine Learning and Cybernetics, 2009 International Conference on*, 2009, pp. 1188-1193.
[135] M. S. Leeson, R. J. Green, M. D. Higgins, and E. L. Hines, "Intelligent Systems Engineering: Optical network applications," in *Transparent Optical Networks (ICTON), 2011 13th International Conference on*, 2011, pp. 1-4.
[136] P. V. K. Borges, N. Conci, and A. Cavallaro, "Video-Based Human Behavior Understanding: A Survey," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 23, pp. 1993-2008, 2013.
[137] C. M. Bishop, *Pattern Recognition And Machine Learning*. Cambridge: Springer, 2006.
[138] S. H. C. Haris, R. B. Ahmad, M. A. H. A. Ghani, and G. M. Waleed, "Packet analysis using packet filtering and traffic monitoring techniques," in *Computer Applications and Industrial Electronics (ICCAIE), International Conference on*, 2010, pp. 271-275.
[139] G. Kramer, M. Gastpar, and P. Gupta, "Cooperative Strategies and Capacity Theorems for Relay Networks," *IEEE Transactions on Information Theory,* vol. 51, pp. 3037-3063, 2005.
[140] C. Yixin and J. Z. Wang, "Support vector learning for fuzzy rule-based classification systems," *IEEE Transactions on Fuzzy Systems,* vol. 11, pp. 716-728, 2003.
[141] J. Gao, B. Sun, and G. Xie, "An improvement algorithm of chaos sequence uniformization," in *2nd International Conference on Industrial and Information Systems*, 2010, pp. 523-526.
[142] J. Haupt, W. U. Bajwa, G. Raz, and R. Nowak, "Toeplitz Compressed Sensing Matrices With Applications to Sparse Channel Estimation," *IEEE Transactions on Information Theory,* vol. 56, pp. 5862-5875, 2010.
[143] M. J. Porsani and T. J. Ulrych, "Levinson-type algorithms for polynomial fitting and for Cholesky and Q factors of Hankel and Vandermonde matrices," *IEEE Transactions on Signal Processing,* vol. 43, pp. 63-70, 1995.
[144] A. Nadas, "Hidden Markov chains, the forward-backward algorithm, and initial statistics," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 31, pp. 504-506, 1983.
[145] T. Jin, Z. Zhou, Q. Song, and W. Chang, "The evidence framework applied to fuzzy hypersphere SVM for UWB SAR landmine detection," in *8th International Conference on Signal Processing*, 2006.
[146] S. Kikuchi and N. Yamanaka, "An expandable time-division circuit switching LSI and network architecture for broadband ISDN," *Selected Areas in Communications, IEEE Journal on,* vol. 14, pp. 328-336, 1996.
[147] S. Vutukury and J. J. Garcia-luna-aceves, "A distributed algorithm for multipath computation," in *Global Telecommunications Conference*, 1999, pp. 1689-1693 vol.3.
[148] J. Cohen and R. Sitver, "A Case Study in Program Transformation: Translation into Polish," *IEEE Transactions on Software Engineering,* vol. SE-5, pp. 593-606, 1979.
[149] A. Silberschatz, "Extending CSP to Allow Dynamic Resource Management," *IEEE Transactions on Software Engineering,* vol. SE-9, pp. 527-531, 1983.
[150] V. Cerf and R. E. Kahn, "A Protocol for Packet Network Intercommunication," *IEEE Transactions on Communications,* vol. 22, pp. 637-648, 1974.
[151] L. Tuong, G. Kuthethoor, C. Hansupichon, P. Sesha, J. Strohm, G. Hadynski*, et al.*, "Reliable User Datagram Protocol for airborne network," in *Military Communications Conference. IEEE*, 2009, pp. 1-6.
[152] M. G. Gouda and M. Schneider, "Maximizable routing metrics," in *Network Protocols. Proceedings. Sixth International Conference on*, 1998, pp. 71-78.

[153] J. Anxiao, M. Schwartz, and J. Bruck, "Correcting Charge-Constrained Errors in the Rank-Modulation Scheme," *IEEE Transactions on Information Theory,* vol. 56, pp. 2112-2120, 2010.
[154] R. E. Cox, "Traffic flow in an exponential delay system with priority categories," *Proceedings of the IEE - Part B: Radio and Electronic Engineering,* vol. 102, pp. 815-818, 1955.
[155] P. Smith and P. Dmochowski, "Exact Blocking Time Statistics for the Erlang Loss Model," *IEEE Wireless Communications Letters,* vol. PP, pp. 1-4, 2013.
[156] M. Kavacky, E. Chromy, and J. Suran, "Evaluation of Erlang models in IP network," in *Emerging eLearning Technologies and Applications, 9th International Conference on*, 2011, pp. 109-113.
[157] M. Mishali and Y. C. Eldar, "From Theory to Practice: Sub-Nyquist Sampling of Sparse Wideband Analog Signals," *IEEE Journal of Selected Topics in Signal Processing,* vol. 4, pp. 375-391, 2010.
[158] O. Aboul-Magd and B. Jamoussi, "QoS and service interworking using constraint route label distribution protocol (CR-LDP)," *Communications Magazine, IEEE,* vol. 39, pp. 134-139, 2001.
[159] A. A. Kist, "Erlang B as a Performance Model for IP Flows," in *IEEE 15th International Conference on Networks*, 2007, pp. 37-41.
[160] C. Shun-Ping, A. Kashper, and K. W. Ross, "Computing approximate blocking probabilities for large loss networks with state-dependent routing,", *IEEE/ACM Transactions on Networking,* vol. 1, pp. 105-115, 1993.
[161] A. Ray, R. M. Angevine, and J. F. Haughney, "Service Access Procedure (SAP) in Support of the WIS Architecture," in *IEEE on Military Communications Conference* 1985, pp. 404-408.
[162] J. Thomas, M. Douglas, R. Watanabe, H. E. Henrikson, M. Z. Iqbal, L. W. Mitchell*, et al.*, "Real Time Data Acquisition for a Time Projection Chamber Using a High Speed DEC-RT11 to Unix UDP-TCP/IP Interface," *IEEE Transactions on Nuclear Science,* vol. 34, pp. 845-848, 1987.
[163] R. Morris and H. T. Kung, "Impact of ATM switching and flow control on TCP performance: measurements on an experimental switch," in *IEEE Global Telecommunications Conference*, 1995, pp. 888-892 vol.2.
[164] A. Salem and J. W. Atwood, "Formal validation of the security properties of AMT's three-way handshake," in *Local Computer Networks (LCN), IEEE 36th Conference on*, 2011, pp. 227-230.
[165] J. Postel, "RFC 793: Transmission control protocol," 1981.
[166] H. Ohba, Y. Yoshida, T. Nakajo, and T. Nagata, "On the Packet-Interleaved Interface Between Packet-Switched Network and Computers," *Communications, IEEE Transactions on,* vol. 22, pp. 1671-1675, 1974.
[167] M. Nesenbergs, "Comparison of the 3-out-of-7 ARQ with Bose-Chaudhuri-Hocquenghem Coding Systems," *IEEE Transactions on Communications Systems,* vol. 11, pp. 202-212, 1963.
[168] D. Chase, P. D. Muellers, and J. K. Wolf, "Application of Code Combining to a Selective-Repeat ARQ Link," in *IEEE on Military Communications Conference* 1985, pp. 247-252.
[169] D. C. Schuurman and D. W. Capson, "Real-time synchronized vision sensors over Ethernet," in *29th Annual IEEE International Conference on Local Computer Networks,.* , 2004, pp. 136-143.
[170] D. D. Clark and W. Fang, "Explicit allocation of best-effort packet delivery service," *IEEE/ACM Transactions on Networking,* vol. 6, pp. 362-373, 1998.
[171] V. Cerf, E. Harslem, J. Heafner, R. Metcalfe, and J. White, "An Experimental Service for Adaptable Data Reconfiguration," *IEEE Transactions on Communications,* vol. 20, pp. 557-564, 1972.
[172] F. A. Tobagi, "Carrier Sense Multiple Access with Message-Based Priority Functions," *IEEE Transactions on Communications,* vol. 30, pp. 185-200, 1982.
[173] WireShark (2013, July 11th) WireShark Available: http://www.wireshark.org/
[174] T. D. Ndousse and S. P. R. Kumar, "PPP extensions for IP/PPP-HDLC over SONET-SDH/WDM," in *IEEE International Conference on Communications*, 1999, pp. 575-580.

[175] J. Baliga, R. W. A. Ayre, K. Hinton, and R. Tucker, "Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport," *Proceedings of the IEEE,* vol. 99, pp. 149-167, 2011.
[176] P. H. Baechtold, M. P. Beakes, P. Buchmann, R. Clauberg, J. F. Ewen, J. F. Gilsdorf*, et al.*, "Single-chip 622-Mb/s SDH/SONET framer, digital cross-connect and add/drop multiplexer solution," *IEEE Journal of Solid-State Circuits,* vol. 36, pp. 74-80, 2001.
[177] M. Luise and R. Reggiannini, "Carrier frequency recovery in all-digital modems for burst-mode transmissions," *IEEE Transactions on Communications,* vol. 43, pp. 1169-1178, 1995.
[178] P. Shakarian, J. Shakarian, and A. Ruef, *Introduction to Cyber-Warfare: A Multidisciplinary Approach*: Syngress Publishing, 2013.
[179] S. Singh and S. Silakari, "An Ensemble Approach for Cyber Attack Detection System: A Generic Framework," in *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), ACIS International Conference* 2013, pp. 79-84.
[180] T. Kiravuo, M. Sarela, and J. Manner, "A Survey of Ethernet LAN Security," *Communications Surveys & Tutorials, IEEE,* vol. 15, pp. 1477-1491, 2013.
[181] Y. Yang and W. Yonggang, "A Software Implementation for a Hybrid Firewall Using Linux Netfilter," in *Second World Congress on Software Engineering (WCSE)*, 2010, pp. 18-21.
[182] I. H. Lim, S. Hong, M. S. Choi, S. J. Lee, T. W. Kim, S. W. Lee*, et al.*, "Security Protocols Against Cyber Attacks in the Distribution Automation System," *Power Delivery, IEEE Transactions on,* vol. 25, pp. 448-455, 2010.
[183] E. Cayirci and R. Ghergherehchi, "Modeling cyber attacks and their effects on decision process," in *Simulation Conference (WSC)*, 2011, pp. 2627-2636.
[184] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," presented at the Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, Seattle, Washington, USA, 1999.
[185] P. N. Ayuso, L. Lefevre, and R. M. Gasca, "hFT-FW: Hybrid Fault-Tolerance for Cluster-Based Stateful Firewalls," in *14th IEEE International Conference on Parallel and Distributed Systems ICPADS*, 2008, pp. 525-532.