10

Performance, Models and Topologies of Common Channel Signalling System Number Seven

R. S. Ramlakan

University of Durban-Westville

Department of Electrical Engineering

Supervisor : Professor S. H. Mneney

To Shabashni.

I would like to thank Professor S. H. Mneney for his help and advice in my MSc studies. I would also like to thank Telkom SA, Ericsson and THRIP for their continuous financial aid during the course of my studies. Special thanks to Dr K. Sandrasegran and N. R. Naidoo as well as my parents for their guidance and motivation.

Declaration

I declare that this project report is my own, except where otherwise acknowledged. It is being submitted for the degree of Master of Science in Engineering in the University of Durban Westville, South Africa. It has not been submitted before for any degree or examination in any other university.

Signed this 17th day of April 1998

....

R. S. Ramlakan

Table of Contents

List of Figures

List of Tables

List of Graphs

List of Simulations

List of Maps

List of Symbols

a - traffic loading by message signal units (MSU) (excluding retransmission);

A_x - availability of element x;

n - number STPs involved;

P_u - error probability of message signal units;

P_x - availability of element x;

Q_a - queueing delay (absence of disturbances).

Q_t - total queueing delay;

R_x - relability of element x;

T_{cs} - message transfer time at STPs;

T_f - emission time of fill-in signal units;

T_L - signalling loop propagation time including processing time in signalling terminal;

T_m - mean emission time of message signal units;

T_{mr} - MTP receiving time;

T_{ms} - MTP sending time;

T_o - overall message tranfer time (presence of disturbances);

T_{oa} - overall message transfer time (absence of disturbance);

T_p - data channel propagation time;

$t_f = \frac{T_f}{T_L}$; $t_L = \frac{T_L}{T_m}$;

$E_2 = 1 + P_u t_L (t_L + 2)$; $E_1 = 1 + P_u t_L$;

List of Acronyms

ADP	- address resolution packet
CCS7	- common channel signalling system no. 7
FCFS	- first come first served
FISU	- fill in signal unit
IN	- intelligent network
ISO	- international standards organisation
MSU	- message signal unit
MSC	- master station controller
MTP	- message transfer part
MTPL1	- MTP layer 1
MTPL2	- MTP layer 2
MTPL3	- MTP layer 3
OSI	- Open Systems Interconnections
PCR	- preventive cyclic retransmission (error correction method)
RAD	- route advertisement (packet)
RL2	- receiving MTPL2
RL3	- receiving MTPL3
SL1	- sending MTPL1
SL2	- sending MTPL2
SL3	- sending MTPL3

SP - signalling point

STP - signalling transfer point

TB - transmission buffer

UP - user part

Abstract

An Intelligent Network (IN) can be visualised as a central computer interconnected to the telecommunication network that allows the creation and deployment of services as consumer demand arises. Messages or information are transferred between the central processor and exchanges by a standardised form of signalling, known as Common Channel Signalling System Number Seven (CCS7).

The thesis focuses on CCS7 performance, models and topologies. A requirement of the research entails the study of the CCS7 performance parameters : signalling delays and signalling network dependability. Signalling delays may be comprised of signalling message transfer delays within the message transfer part and queueing delays. Signalling network dependability usually include availability, reliability, maintainability and network robustness.

For the purpose of modelling, the decomposition of the CCS7 message transfer part into subsystems is essential. A generic model for the message transfer part was used to implement certain functionalities of CCS7 in OPNET. OPNET (Optimised Network Engineering Tools) is a comprehensive engineering system capable of simulating large communication networks with detailed protocol modelling and performance analysis. The simulation software (OPNET) and performance parameters were used to analyse CCS7 networks resulting in the selection of a particular topology for a given region.

The network structures analysed include the mesh topology, standard quad topology, hierarchical topology and Telstra's new topology. It was found that the mesh structure had the best simulated and availability/reliability results but was impractical for large networks due to the cost implications. This cost factor led to the selection of a hierarchical signalling system for Kwa-Zulu Natal.

1. Introduction

1.1 Background information

The communication among the physical elements of an intelligent network (IN) can be accomplished by the standardised Common Channel Signalling System Number 7 (CCS7). CCS7 entails separate management of the signalling network and the voice/data network whereas in the past, signalling management was integrated into call routing management.

Figure 1.1 shows a CCS7 network consisting of signalling points (SPs), signalling transfer points (STPs) and signalling links. Each SP is associated with a corresponding local or transit exchange in the circuit switched network (this correspondence can be represented by the thick double arrow lines in between the SPs and exchanges), whereas STPs have no counterpart in that network. From the functional point of view, the SPs can further be subdivided into SPs associated with local exchanges (origination/destination SPs) and SPs associated with transit exchanges (intermediate SPs) [24].

The ITU-T (International Telecommunications Union) or CCITT (Comite Consultatif International Telegraphique et Telephonique) began the standardisation of CCS7 in 1976. The goal of ITU-T CCS7 was to offer an internationally standardised signalling system for the interconnection of computer controlled telephone exchanges. The system was to be suitable for the handling of call build up, management information and maintenance messages - including those for what were then future systems, such as ISDN and circuit-switched data networks.

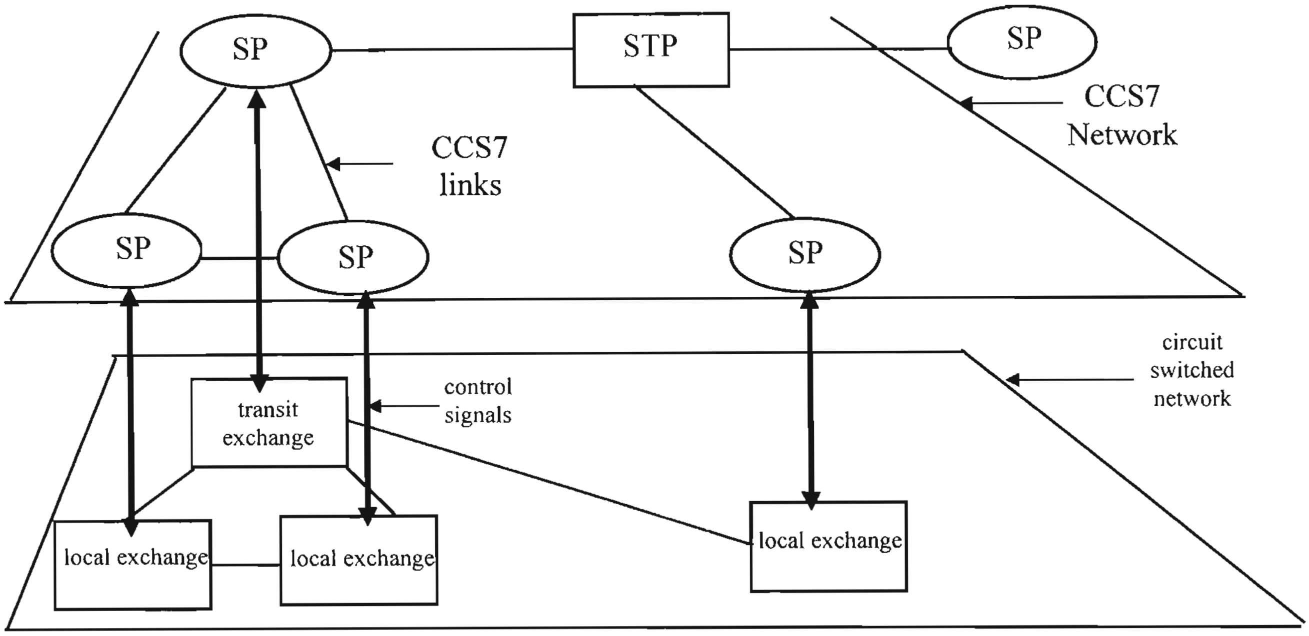


Figure 1.1 : Relation between the CCS7 network and the circuit switched network.

Many countries have installed systems using CCS7, although these are often national versions. The system is optimised for the use of 64 kbit/s channels for signalling, although it can operate on analogue channels with lower bit rates. Satellite links can also be used [22].

1.2 Protocol architecture

The CCS7 protocol is based on the Open Systems Interconnections (OSI) model and is divided into different layers corresponding to functionality (in the case of CCS7, these layers are called levels). While the OSI model consists of seven different layers, the CCS7 standard uses only four levels [28]. The functions carried out by these four levels correspond to the OSI model's seven layers (see figure 1.2).

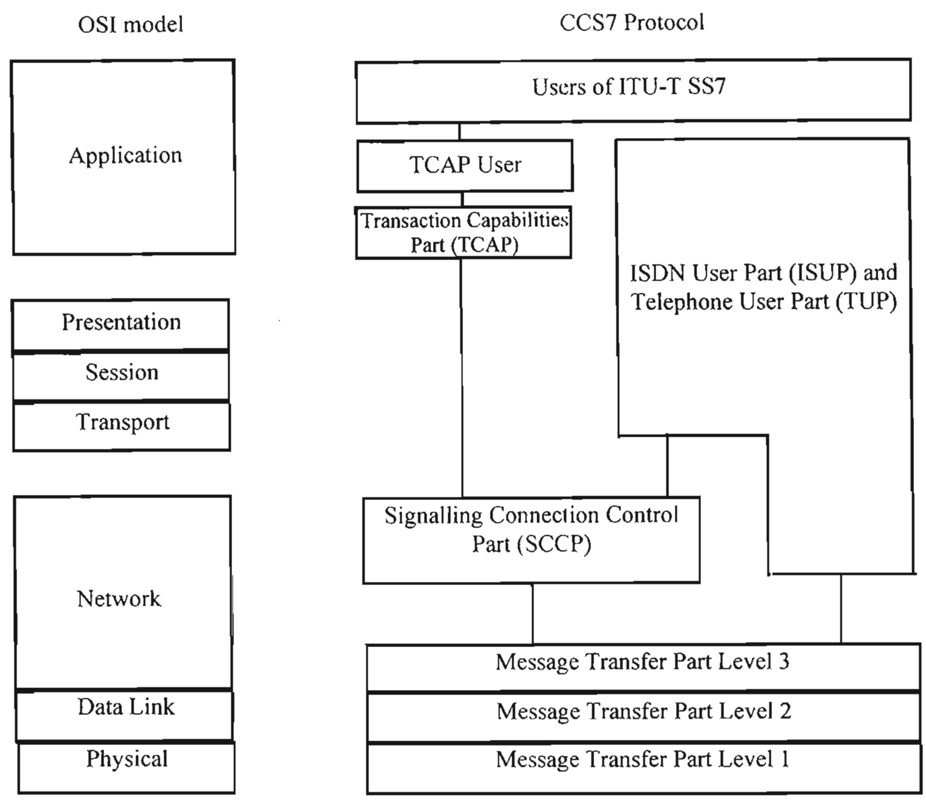


Figure 1.2 : Relation between the OSI model and the CCS7 protocol

The OSI physical layer, data link layer and parts of the network layer are contained in the Message Transfer Part (MTP). The MTP is responsible for the reliable operation of signalling networks and consists of the lower three CCS7 protocol levels : MTP level 1 (MTPL1), MTP level 2 (MTPL2) and the MTP level 3 (MTPL3).

The MTPL1 defines the physical, electrical and functional characteristics of a signalling data link and a means of link access. In a digital environment, 64 kbits/s digital paths will normally be used for the signalling digital links. The signalling link may be accessed via a switching function, providing a potential for automatic reconfiguration of signalling links. Other types of data links, such as analogue links with modems, can also be used [26].

The MTPL2 defines the functions and procedures for, and relating to, the transfer of signalling messages over one individual signalling data link. The MTPL2 ensures the reliable sequenced delivery of data packets across the MTPL1 by the following signalling link functions :-

- delimitation of signal units by means of flags;
- flag imitation prevention by bit stuffing;
- error detection by means of check bits in each signal unit;
- error correction by retransmission and signal unit sequence control by means of explicit sequence numbers in each signal unit and explicit continuous acknowledgements;
- signalling link failure detection by means of signal unit error rate monitoring and signalling link recovery by means of special procedures.

The MTPL3 define transport functions and procedures that are common to the operation of individual signalling links. These functions and procedures fall into two major categories :-

- The message handling functions which can be divided into the message routing, discrimination and distribution functions. These are functions that, at the actual transfer of a message, direct the message to the proper signalling link or User Part (UP).

- The signalling network management function can be divided into the traffic management function, the link management function and the route management function. The network management function provides reconfiguration of the signalling network in the event of failures and controls traffic in the case of congestion.

The CCS7 level 4 consists of the different User Parts (UPs). Each UP defines the functions and procedures of the signalling system that are particular to a certain type of user in the system. The users for which communication functions are defined can occur within the signalling system and external to the signalling system. One of the users within CCS7 is the telephone functions with their corresponding Telephone User Part (TUP) while an example of an external user is the use of the signalling system for transfer of information for some management or maintenance function.

This thesis will concentrate on the MTP, since it is the responsibility of the MTP to transfer messages between different UPs. Signalling system performance depends on the capability of the MTP to transport messages in a defined manner. Lengthy delays that occur as messages are transferred between different users affect the network response time and therefore increase the service set-up time.

1.3 Report overview

The objective of this research is to recommend a CCS7 network topology for a specific region, e.g. Kwa-Zulu Natal. The selection of the appropriate network configuration requires the identification of necessary CCS7 performance parameters, the use of models to implement CCS7 functionality in a telecommunication software package (OPNET) and the study of common topologies associated with the signalling system.

Performance parameters necessary in network selection are defined in chapter 2. The parameters to be discussed can be classified as signalling delay and signalling network dependability. Signalling delays consists of message transfer delays, queueing delays and overall message transfer delays while network dependability includes availability, reliability, robustness and survivability.

Chapter 3 entails the description of the model for the MTP. This section also shows the model implementation in OPNET where the necessary processes and functions will be outlined and the signalling system operation validated through adequate simulations. For the simulation, both the quad structure and a simple SP-SP relation will be investigated.

Analysed and simulated results pertaining to different topologies can be found in chapter 4. The quality of service parameters and the simulations from OPNET discussed in chapters 2 and 3 will be used to determine the most feasible network. This section will be followed by a conclusion with possible recommendations.

2. Performance Parameters and Objectives of CCS7

2.1 Introduction

Control information necessary for call handling and network management can be handled by a signalling network; the Common Channel Signalling System Number Seven (CCS7). Information has to be transported efficiently, since all delays resulting from the network affect the response times of call handling and network management actions[1]. This in turn affects the performance parameters of the signalling network.

Performance parameters necessary in network modelling include: the maximum delay and load permitted in a CCS7 section and the entire network; the use of load sharing in signalling links and the connectivity criteria (depending on demand); and the failure of one or more CCS7 elements (section or node)[2]. It should be noted that modelling only creates an overall view of the network operation. In practice, modelling can only approximate the influences that the advent of new services have on existing CCS7 networks.

It is assumed that a signalling performance which satisfies the most stringent User Part (UP) requirement will also meet those of future users, [3] and [4]. Signalling system performance is understood to be the capability of the Message Transfer Part (MTP) to transfer messages of variable length for different users in a defined manner. In order to achieve a proper signalling performance, the following groups of performance parameters will have to be taken into account :-

- The first group covers signalling delays where emphasis is placed on message transfer delays, queueing delays and overall message transfer delays;
- The second group focuses on the attributes of signalling network dependability : availability, reliability, maintainability and survivability;

The above groups coupled with load sharing and the network management functions enable the MTP to transfer messages in such a way that the signalling requirements of all users are met and satisfactory overall system performance is achieved.

2.2 Signalling delays

2.2.1 Signalling message transfer delays

Signalling message transfer delay affects the call setup delay and the network response time to service requests. The capabilities of individual components and the signalling network structure also influence the constraints on maximum signalling delays allowed [3].

Within a signalling relation, the Message Transfer Part (MTP) transfer messages from the originating User Part to the User Part of destination, using several signalling paths. In [4], the message transfer times in each component of signalling relation are defined :-

- **The Message Transfer Part sending time** (T_{ms}) at the point of origin starts when the last bit of the message has left the UP and ends when the last bit of the signal unit enters the signalling data link (see figure 2.1). The outgoing link delays (T_{od}) for varying message lengths are given in table 2.1, [4].

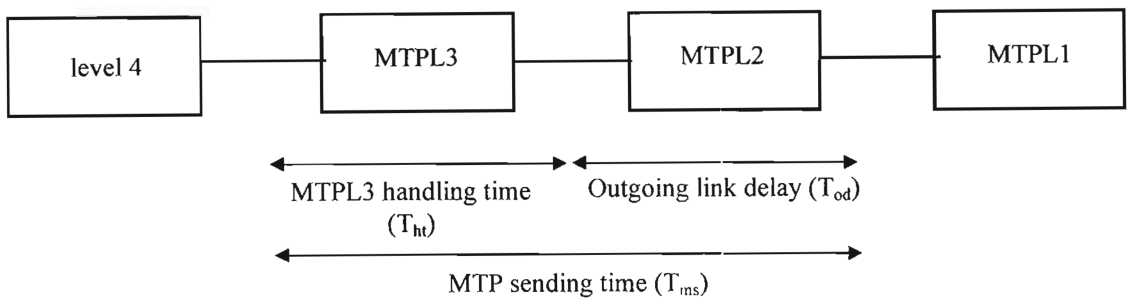


Figure 2.1 : Functional diagram indicating the components of the MTP sending time (T_{ms})

Table 2.1 shows the values for the outgoing link delays. T_L refers to the signalling loop delay and is dependent on the length of the link. T_L indicates the period of time elapsed for a message signal unit (MSU) to be emitted until that MSU has been positively acknowledged by the receiving terminal. The table indicates the link delay for different values of loading in, the presence or absence of disturbances (errors), the mean link delay and where 95% of these link delays occurred.

a [Erl]	T_L [ms]	Disturbance	Value	Outgoing link delay (ms)				
				MSU – Length (Bytes)				
				15	23	50	140	279
0.2	30	No	Mean	2.7	4.0	8.3	21.5	39.6
			95%	9.3	14.0	30.1	66.0	61.5
		Yes	Mean	2.8	4.1	8.4	21.9	40.4
			95%	10.8	15.4	31.4	68.0	64.8
	600	No	Mean	2.7	4.0	8.3	21.5	39.6
			95%	9.3	14.0	30.1	66.0	61.5
		Yes	Mean	29.6	31.2	36.9	55.0	80.3
			95%	248.4	254.2	275.0	329.4	363.8
0.4	30	No	Mean	3.5	5.2	10.8	27.6	46.9
			95%	12.2	18.6	40.0	88.7	87.1
		Yes	Mean	3.8	5.4	11.2	28.3	48.1
			95%	14.3	20.4	41.7	91.4	91.3
	600	No	Mean	3.5	5.2	10.8	27.6	46.9
			95%	12.2	18.6	40.0	88.7	87.1
		Yes	Mean	86.3	88.5	96.9	121.9	152.0
			95%	490.1	496.4	521.2	586.6	626.9

A Traffic loading by MSUs (excluding retransmission)

T_L Signalling link loop propagation time including processing time in signalling terminal mean bit error probability $< 10^{-5}$ in the presence of disturbance.

Table 2.1 : Outgoing link delays (T_{od}).

- **The message transfer time at signalling transfer** (T_{cs}) points is the period which starts when the last bit of the signal unit leaves the incoming signalling data link and ends when the last bit of the signal unit enters the outgoing signalling data link. (refer to figure 2.2). The STP processor handling time (T_{ph}) can be seen in table 2.2, [4].

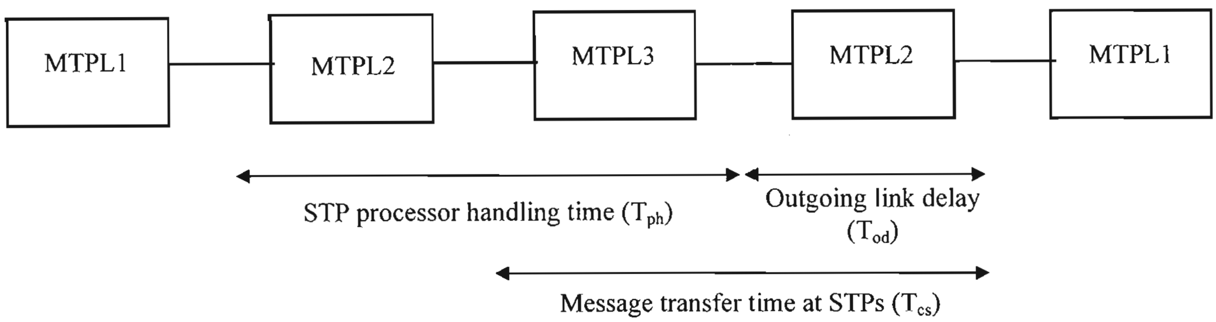


Figure 2.2 : Functional diagram indicating the components of the message transfer time at STPs

Table 2.2 indicates the STP processor handling time in milliseconds at normal loading (approximately 0.2 erlang) and 30% above that. The processor delay time can be given as a mean value or where 95% of these values occurred. The table simply indicates that the mean and 95% of the processor handling times (T_{ph}) increase as the loading or the length of MSUs increase.

Processor load	Delay value	Mean message SU length			
		23 bytes	50 bytes	140 bytes	279 bytes (Note)
Normal	Mean	19	22	33	55
	95%	35	40	50	75
+ 30%	Mean	60	70	100	160
	95%	120	140	200	320
NOTE – The MSU size is fixed in this case at 279 bytes.					

Table 2.2 : STP processor handling time (T_{ph}).

- **The MTP receiving time** (T_{mr}) at the point of destination is the period which starts when the last bit of the signal unit leaves the signalling data link and ends when the last bit of the message has entered the UP (see figure 2.3).

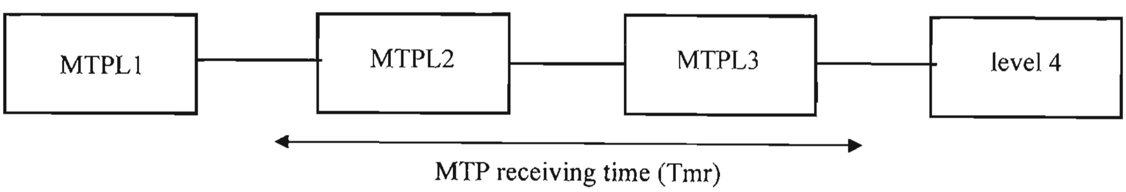


Figure 2.3 : Functional diagram of the MTP receiving time (T_{mr})

- **The data channel propagation time** (T_p) is the period which starts when the last bit of the signal unit has entered the data channel at the sending side and ends when the last bit of the signal unit leaves the data channel at the receiving end, irrespective of whether the signal unit is disturbed or not (refer to figure 2.4). T_p depends on transmission speed, the distance between nodes, repeater spacing and the delays in the repeaters. Transmission speed and repeater delay depend on the type of medium (wire, fibre optic cable, terrestrial radio and satellite systems) used to transmit the messages (refer to table 2.3 and graph 2.1, [4]).

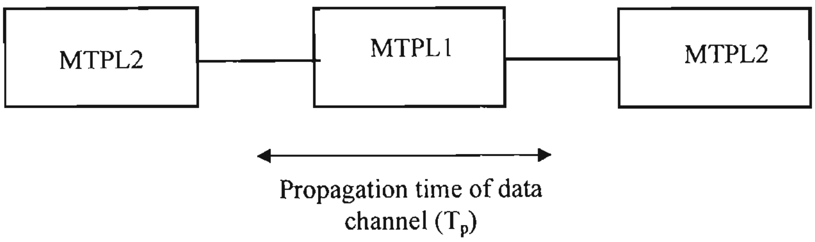
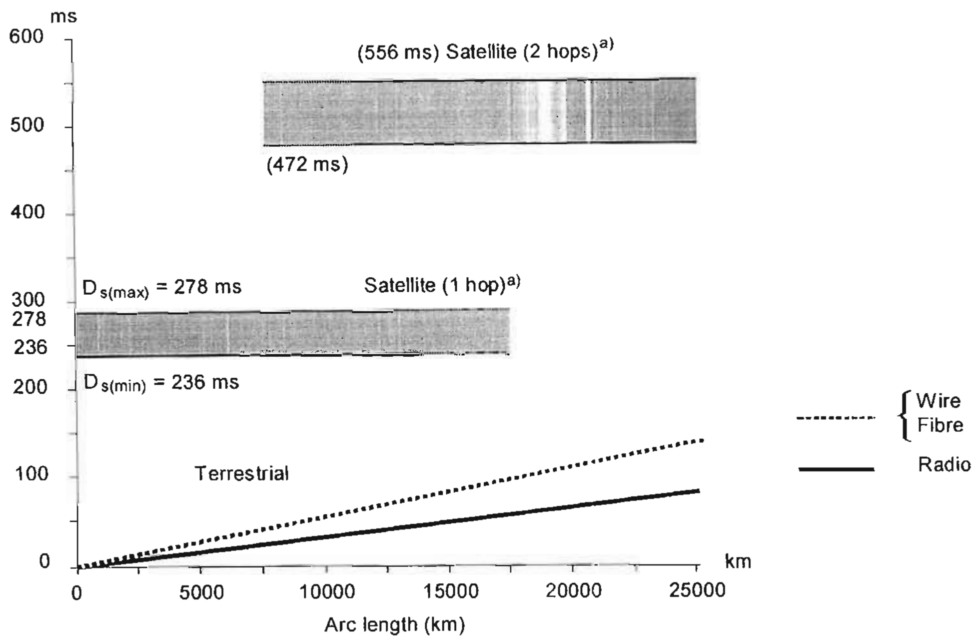


Figure 2.4 : Functional diagram for propagation time (T_p)

Arc length (km)	Delay terrestrial (ms)		
	Wire	Fibre	Radio
500	2.4	2.50	1.7
1 000	4.8	5.0	3.3
2 000	9.6	10.0	16.6
5 000	24.0	25.0	16.5
10 000	48.0	50.0	33.0
15 000	72.0	75.0	49.5
17 737	85.1	88.7	58.5
20 000	96.0	100.0	66.0
25 000	120.0	125.0	82.5

Table 2.3 : Calculated transmission delays for various call distances



a) Delay varies with distances of each earth station from point on the Earth over which the satellite is located.

Graph 2.1 : Mean delay (ms) vs distance (km)

2.2.2 Queueing delays

The MTP handles messages from different User Parts on a time-shared basis. With time-sharing, signalling delay occurs when it is necessary to process more than one message in a given interval of time. When this occurs, a queue is built up from which messages are transmitted in order of their times of arrival.

The formulae summarised in the ITU-T recommendations give expressions for the mean message queueing delay for in the MTPL2 under certain simplifying assumptions [5], [6]. There are two different types of queueing delays : queueing delays in the absence of disturbances and the queueing delay in the presence of disturbances. In the presence of disturbances, transmission errors of message signal units (MSUs) are random and statistically independent of one another. The additional delay caused by the retransmission of an erroneous signal unit is considered as part of the waiting time of the original signal unit[4]. In CCS7, two forms of error correction are defined : the basic error correction method and the preventive cyclic retransmission (PCR) method. The PCR method will not be considered since it is only useful for satellite links or terrestrial links with long propagation delays.

The basic method is a compelled, positive/negative acknowledgement, retransmission error correction system. Signalling information is transferred in CCS7 using a specific message format called message signal units (MSUs). MSUs are transmitted on a first come first served (FCFS) basis, and a MSU that has been transmitted is retained in the retransmit buffer until a positive acknowledgement for that MSU is received. If a negative acknowledgement is received, the transmission of new MSUs are interrupted. The negatively acknowledged MSU together with all those MSUs transmitted after it are retransmitted once in the order they were first transmitted. If there are no new MSUs to be transmitted and no MSUs are being retransmitted, fill-in signal units (FISUs), which are 6 octets long, are transmitted [5].

The MTPL2 queueing delay for the basic error correction method is defined as the time elapsed between the arrival of the first bit of a MSU in the transmission buffer (TB) and the emission of the first bit onto the data channel [38]. The model is based on the first come first serve (FCFS) M/G/1 queue where the interarrival time distribution is Poisson [M], the service time distribution is general [G], the number of servers is one [1], and priority is given to MSUs over fill-in signal units (FISUs). The signalling link loop propagation time is assumed constant and includes the process time in signalling terminals.

The notations and factors required for calculating the queueing delay can be defined as follows:

- Q_a - mean queueing delay in the absence of disturbances;
- Q_t - mean total queueing delay;
- σ_a^2 - variance of queueing delay in the absence of disturbances;
- σ_t^2 - variance of total queueing delay;
- a - traffic loading by message signal units (MSU), excluding retransmission;
- T_m - mean emission time of message signal units;
- T_f - emission time of fill-in signal units;
- T_L - signalling loop propagation time including processing time in signalling terminal;
- P_u - error probability of message signal units;

$$k_1 = \frac{2nd \ \ moment \ \ of \ \ MSUs \ \ emission \ \ time}{T_m^2}$$

$$k_2 = \frac{3rd \ \ moment \ \ of \ \ MSUs \ \ emission \ \ time}{T_m^3}$$

The second and third moments of the mean emission time can be obtained from [4]. The parameters defined in the formulae used in the queueing formula is as follows:

$$t_f = \frac{T_f}{T_L} \qquad (2.1)$$

$$t_L = \frac{T_L}{T_m} \quad (2.2)$$

$$E_1 = 1 + P_u t_L \quad (2.3)$$

$$E_2 = 1 + P_u t_L (t_L + 2) \quad (2.4)$$

$$E_3 = k_2 + P_u t_L (t_L^2 + 2t_L + 3k_1) \quad (2.5)$$

The MTPL2 mean queueing delay and the variance can be given in table 2.4. For an analysis of the basic error correction method, Turbo C++ was used to obtain the mean MTPL2 queueing delay for varying signalling traffic loads. The traffic model used in this study was obtained from the ITU-T recommendations. It should be noted that these values were based on the telephone UP (TUP) messages. For other services such as that provided by the ISDN UP, the mean message length would increase and thus the mean queueing delay would increase. The following values were obtained from the recommendations :

- T_L = 3.2 ms for a 100 km length of link;
- the mean MSU and FISU length are 120 bits and 48 bits respectively;
- the data rate of links are 64 kbits/s.

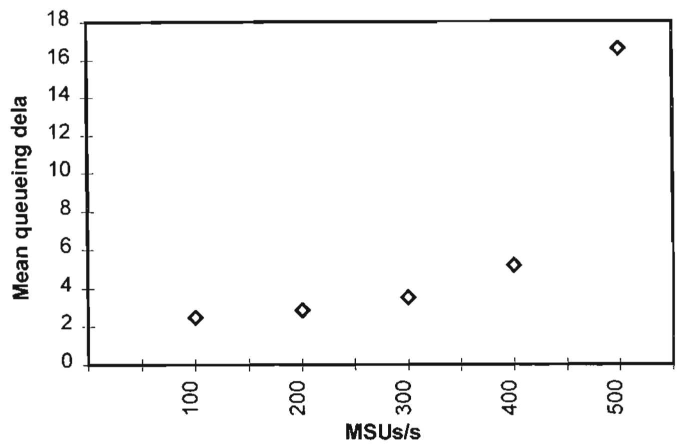
In addition, the probability of MSUs being in error (P_u) was assumed to be zero. Using the above data, the values in table 2.5 can be obtained. The software programme used to calculate these delays are given in appendix B. These values are also presented in graph 2.2. Since the maximum number of MSUs that can be transmitted in 1s is equivalent 533 MSUs, the mean queueing delay rises sharply as the loading of MSUs reaches this limit, i.e. a link does not build up any appreciable MTPL2 queue unless the link approaches full utilisation (1 Erlang of traffic).

Error correction method	Disturbance	Mean Q	Variance σ^2
Basic	Absence	$\frac{Q_a}{T_m} = \frac{t_f}{2} + \frac{ak_1}{2(1-a)}$	$\frac{\sigma_a^2}{T_m^2} = \frac{t_f^2}{12} + \frac{a[4k_2 - (4k_2 - 3k_1^2)a]}{12(1-a)^2}$
Basic	Presence	$\frac{Q_l}{T_m} = \frac{t_f}{2} + \frac{aE_2}{2(1-aE_1)} + E_1 - 1$	$\frac{\sigma_l^2}{T_m^2} = \frac{t_f^2}{12} + \frac{a[4E_3 - (4E_1E_3 - 3E_2^2)a]}{12(1-aE_1)^2} + Pu(1-P_u)\,t_L^2$

Table 2.4 : Statistics for the basic error correction method

No. MSUs/s	T_m (ms)	no. bits MSUs	no. FISUs	T_f (ms)	t_f	k_1	a (loading)	Q_a (ms)
100	0.01	12000	1083	0.0009	0.0923	1.7393	0.1876	2.47
200	0.005	24000	833	0.0012	0.24	1.4918	0.3752	2.84
300	0.0033	36000	583	0.0017	0.5143	1.2362	0.5629	3.51
400	0.0025	48000	333	0.003	1.2	0.9789	0.7505	5.18
500	0.002	60000	83	0.012	6	0.6970	0.9381	16.56

Table 2.5 : Calculated values for mean queueing delay



Graph 2.2 : Mean queueing delay vs. loading

2.2.3 Overall message transfer times

The overall message transfer time T_o, starts when the message has left the user part(level 4) at the point of origin and ends when the message has entered the user part(level 4) at the point of destination. T_o refers to a signalling relation and can be defined in either the absence or presence of disturbances (see equation 2.6a and 2.6b).

In the absence of disturbances :

$$T_{oa} = T_{ms} + \sum_{i=1}^{n+1} T_{p.i} + \sum_{i=1}^{n} T_{cs.i} + T_{mr} \qquad (2.6a)$$

In the presence of disturbances :

$$T_o = T_{oa} + \sum (Q_t - Q_a) \qquad (2.6b)$$

T_{oa} - overall message transfer time (absence of disturbance);

T_{ms} - MTP sending time;

T_{mr} - MTP receiving time;

T_{cs} - message transfer time at STPs;

n - number STPs involved;

T_p - data channel propagation time;

T_o - overall message transfer time (in the presence of disturbances);

Q_t - total queueing delay;

Q_a - queueing delay (in the absence of disturbances).

An estimation of the telephone user part (TUP) message delays at a STP can be obtained from table 2.6, [4]. The STP delays, the data channel propagation time (see table 2.3 for values for T_p), the queueing delay (see tables 2.4 and 2.5 - $Q_a = Q_t$ for no errors) and the MTP sending/receiving times (T_{ms}/T_{mr}) give an approximation for the overall message transfer time (T_{oa}). The end to end (ETE) delays for 50% and 95% of connections are given in Table 2.7 for various combinations of large-size and average-size countries. Average signalling point and STP delays at normal loading are assumed [14].

STP signalling traffic load	(T_{cs}) ms	
	Mean	95%
Normal	20	40
+ 15%	40	80
+ 30%	100	200
NOTE – The values in the table were determined based on TUP messages.		

Table 2.6 : Estimation of STP times (T_{cs})

Country size	Percent of connections	Delay (ms)	
		Message type	
		Simple (e.g. answer)	Processing intensive (e.g. IAM)
Large-size	50%	390	600
	95%	520	800
Average-size	50%	260	400
	95%	390	600
NOTES 1 The delay does not include any delay for the International Switching Centre in the country, which is included in the international component.			

Table 2.7 : Estimation of end to end (ETE) delays.

Call set-up time is increased by both message transfer and queueing delays. Delays that occur as messages are transferred between different user parts affect network response times. The number of signalling points and signalling transfer points should therefore be kept to a minimum in order to reduce delays (see table 2.8). In table 2.8, a large size country can be related to the USA while a medium size country can be represented by Italy. In both these countries, signalling point delay forms the largest component of signalling delays[14],[15].

Country size (Note 1)	Percent of connections	Number of STPs	Number of signalling points
Large-size	50%	3	3
	95%	4	4
Average-size	50%	2	2
	95%	3	3
NOTES 1. The values in this table are provisional. (A higher number of signalling points and/or STPs might be included in a national network, e.g. in the case that a two-level hierarchical signalling network is adopted. This matter is for further study.)			

Table 2.8 : Estimation of the number of SPs and STPs used in a signalling connection

2.3 Signalling network dependability

The CCS7 network is the lifeline of the whole voice network in modern communication networks. It provides the signalling for thousands of calls. One CCS7 link is capable of carrying the signalling for more than nine hundred voice circuits. Therefore if one of the signalling links in the signalling network is down, nine hundred voice calls are lost[7].

In [8] and [9], dependability is defined as call loss, availability/unavailability of routes, maintainability and network robustness or survivability. Call loss is assessed on the basis of MSU loss and specifications places an objective of 1 in 10^7 on loss of messages. Call losses arise from routing data errors, network element failure and route congestion. This section will focus on availability, reliability, maintainability and survivability of CCS7 networks..

2.3.1 Availability

Availability may be defined as the ability of an item to be in a state to perform a required function at a given instant of time or at any instant of time within a given time interval, assuming that the external resources, if required, are provided [9].

The availability of signalling links may be improved by providing diverse extra capacity to handle the load of any failed component. Signalling links are usually engineered to handle twice their normal load. The amount of redundant capacity depends on the network architecture [5]. The required redundancy may be achieved by different combinations of [3] :-

- redundancy in signalling terminal devices;
- redundancy of signalling links within a link set;
- redundancy in signalling links for each destination.

The unavailability of a signalling route set(end-to-end) should not exceed a total of 10 minutes per year [4]. The end-to-end unavailability may result from one or more of the following[8]:-

- combined linkset failure;
- duplex STP failure;
- signalling end point CCS7 interface subsystem failure;
- routing data administration error (procedural error);
- signalling network management failure (software error).

2.3.1.1 Calculating availability

In [10], an analytical method clearly defines a means of calculating the availability of a signalling network infrastructure, given the statistical failure and restoration characteristics of the individual hardware elements in the network. Procedures such as changeover, transfer-prohibited and transfer-restricted are not considered.

The statistical failure and restoration characteristics may be given by MTBF (mean time between failures and MTTR (mean time to restoration) in equation 2.7. The availability is given as :

$$P_A = A_A = \frac{MTBF}{MTBF + MTTR} \tag{2.7}$$

The unavailability is given as :

$$P_U = A_U = \frac{MTTR}{MTBF + MTTR} \tag{2.8}$$

It should be noted that these equations are for the single element analysis. The availability of signalling routes and route sets can be extended from equation 2.7. The signalling routes and route set are defined from the standard quad structure in figure 2.1. For simplicity, only three signalling routes are defined for the route set:-

- Route X = A - 1 - C - 2 - D - 3 - B;
- Route Y = A - 1 - C - 7 - E - 4 - B;
- Route Z = A - 6 - F - 5 - E - 4 - B.

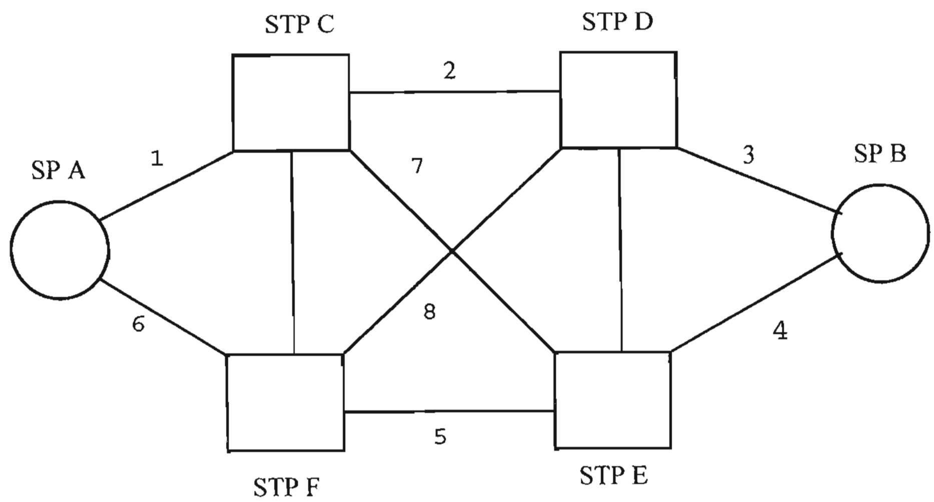


Figure 2.1 : Standard quad structure.

The calculation of the availability of a single signalling route is based on the knowledge that the failure events of the individual network elements are random and independent of one another :

$$P(X) = P(A\&1\&C\&2\&D\&3\&B) = P(A).P(1).P(C).P(2).P(D).P(3).P(B) \qquad (2.9)$$

Similar expressions can be obtained for routes Y and Z.

The availability of the signalling route set is obtained by observing that either route X or Y or Z must be available for the route set to be available. The occurrence of events X, Y or Z are, however, not independent of each other and imply the occurrence of each other. As a result the general relationship for determining the probability of the union of three events has to be used to calculate the availability of the signalling route set. The availability of the route set can be given as :

$$P(X+Y+Z) = P(X) + P(Y) + P(Z) - P(X\&Y) - P(X\&Z) - P(Y\&Z) + P(X\&Y\&Z) \quad (2.10)$$

The expressions used in equation 2.10 can be found in equation 2.11. It should be noted that the availability expressions such as P[Y/X] is computed by multiplying all those availabilities that appear in route Y but not in route X, since Y and X are independent routes. For P[Y/X], the only availabilities that are unique to route X are $A_7A_EA_5$. The unavailability is simply stated as : $P_U = 1 - P_A$.

$$\begin{aligned}
&P(X) = A_X \\
&P(Y) = A_Y \\
&P(Z) = A_Z \\
&P(X\&Y) = P[X]P[Y/X] = A_XA_7A_EA_4 \\
&P(X\&Z) = P[X]P[Z/X] = A_XA_6A_FA_5A_EA_4 \\
&P(Y\&Z) = P[Y]P[Z/Y] = A_YA_6A_FA_5 \\
&P(X\&Y\&Z) = P[X]P[Y/X]P[Z/X,Y] \\
&\qquad = A_XA_7A_EA_4A_6A_FA_5
\end{aligned} \quad (2.11)$$

In [11], the unavailability for a signalling link is expressed in terms of the duration of link operation (M) and the duration of link unavailability for any reason (N) :

$$U_{SL} = \frac{M}{M+N} \quad (2.12)$$

2.3.2 Reliability

Reliability may be defined as the ability of an item to perform its required function under given conditions for a given time interval [9].

The function of signalling networks is to exchange messages for call setup. If messages cannot be transferred then calls cannot be made. The major threats to network reliability can be categorised as follows, [12] :-

- Equipment configuration changes;
- Software errors;
- Routing data changes.

In [12], reliability is improved by dividing networks into regions, with each region having its own management. A routing data error may effect communication in a region but not outside that region.

2.3.2.1 Calculating reliability

In [10], the reliability is expressed as the probability that the time to failure of an element will exceed a time t :

$$R(t) = P(X > t) = \exp(\frac{-t}{MTBF}) \tag{2.13}$$

For the calculation of reliability, figure 2.1 is again considered and the following routes were defined :-

- Route X = A - 1 - C - 2 - D - 3 - B;
- Route Y = A - 1 - C - 7 - E - 4 - B;
- Route Z = A - 6 - F - 5 - E - 4 - B.

The reliability of single signalling route may be expressed as :

$$R_X(t) = R_X(t)R_1(t)R_C(t)R_2(t)R_D(t)R_3(t)R_B(t) \tag{2.14}$$

Similar expressions can be obtained for routes Y and Z. The expression for the reliability of a route set can be obtained in the same way as the expression for the availability of the same route set (see equations 2.10 and 2.11). The reliability of the signalling route set is thus expressed as follows :

$$\begin{aligned} R_{set} &= R_X(t)+R_Y(t)+R_Z(t)-R_X(t)R_7(t)R_E(t)R_4(t) \\ &-R_X(t)R_6(t)R_F(t)R_5(t)R_E(t)R_4(t)-R_Y(t)R_6(t)R_F(t)R_5(t) \\ &+R_X(t)R_7(t)R_E(t)R_4(t)R_6(t)R_F(t)R_5(t) \end{aligned} \tag{2.15}$$

2.3.3 Maintainability

In [9], maintainability is defined as the ability of an item under stated conditions of use, to be retained in, or restored to, a state in which it can perform a required function, when maintenance is performed under given conditions, and using stated procedures and resources.

An alternative definition can be found in [8]. Maintainability encompasses several measures of service quality, including Mean Time To Restore (MTTR), and Maintenance Procedural Error Rate (MPER). MTTR is the mean time to restore signalling capability that has been affected due to one or more signalling route failures. MPER is the frequency of failures due to unsuccessful or erroneous execution of procedures by maintenance personnel. Maintainability can be affected by one or more of the following :-

- Timeliness of trouble location (latency to fault detection)

- Ease of Fault Treatment (complexity of Reconfiguration/Repair)

- Accuracy of Fault Treatment following the onset of a routing data error, element failure, or connectivity failure in the network.

2.3.4 Network robustness and survivability

In [9], survivability is described as the ability of an item to perform a required function at a given instant in time after a specified subset of components of the item become unavailable. Survivability may be improved by architectural diversity, i.e. CCS7 network elements are arranged in quads and pairs to achieve a high level of redundancy [13].

Robustness is intended to quantify a network's vulnerability to large failure scenarios (e.g. large number of signalling points isolated; network-wide service unavailability; 5000 or more calls blocked/lost by a single event). Single events of this kind have the potential to destroy customer confidence or create life-threatening situations due to a loss of emergency service.

In the event of link and nodal failures, messages are routed to their destinations by control procedures that reconfigure networks. These control procedures may be classified into signalling network management functions and the load sharing of traffic.

In load sharing, message routing principles split the entire signalling traffic into partitions. The distribution of traffic provides a useful means of controlling the load of different sections of a signalling network. Load sharing can occur between links belonging to the same link set or links not belonging to the same link set. Load sharing is always related to the same routing for a given destination.

In load sharing between links belonging to the same link set, the traffic flow carried by a link set is shared. An example of such a case is given by a link set directly interconnecting the origination points and destination points in the associated mode of operation. In the latter case, traffic relating to a given destination is shared between signalling links not belonging to the same linkset. The load sharing rule used for a particular signalling relation may or may not apply to all signalling relations which use one of links involved [17].

The number of signalling links used to share the load of a given flow of signalling traffic depends on :-

- the total traffic load;
- the availability of links;
- the required availability of the path between the two signalling points concerned;
- the bit rates of the signalling links.

Load sharing requires at least two links for all bit rates, but more may be needed at lower bit rates. When two links are used, each of them should be able to carry the total signalling traffic in case one link fails [15].

The signalling network management functions consists of the signalling link, traffic and route management functions. The task of the signalling link management function is to control the locally connected link sets and to supply local link availability status information to the traffic management function[18] (see 4.3). The signalling link management has the ability to restore failed links and to activate/deactivate links.

The signalling traffic management performs the diversion and flow control of traffic in response to link/node failures. In the case of link/route unavailability, routing information are based on status information received from the signalling link and signalling route management functions. The objective is to maintain connectivity at all times[18]. Changeover is a signalling traffic management function for the diversion of traffic from a failed link to one or more other links . Changeover avoids message loss, duplication or mis-sequencing. The changeback procedure is then used to divert traffic back to the link when it becomes available again [18].

The purpose of the signalling route management is to ensure that SPs are reliably informed about the availability of routes in the network. The unavailability of a signalling route is communicated to the source SP of a signalling route set by means of procedures and explicit messages : transfer prohibited (TFP), transfer available (TFA) and transfer restricted (TFR).

The TFP message is used toward adjacent nodes for traffic diversion from unavailable nodes. The local route management function generates the TFP message for one or more adjacent nodes. At nodes that receive the TFP messages, the local traffic management function exercises a forced rerouting procedure to reroute traffic away from the unavailable route. When a route is again available, the transfer available (TFA) procedure is followed involving the transmission of TFA messages to adjacent nodes, where the local traffic management function may invoke a controlled rerouting procedure to reuse the now available route[18].

A TFR message is used towards one or more adjacent nodes for traffic diversion to alternative routes (if possible) in response to traffic congestion. The local route management function uses a TFR message to carry the indication to adjacent nodes. At a node in receipt of a TFR message, the local management procedure invokes a controlled rerouting procedure, if an alternative link set is available, to redirect traffic away from that route. When a route is again available, the TFA procedure is followed involving the transmission of TFA messages to adjacent nodes, where the local traffic management function may invoke a forced rerouting procedure to reuse the now available route.

3. A Model for the MTP

3.1 Introduction

The model for a signalling link directly interconnecting signalling points (SPs) is slightly different from the case where messages are routed over one or more signalling transfer points, i.e. quasi-associated mode of signalling. The sending and receiving message transfer part (MTP) layers 1 and 2 remains the same for both models whereas the signalling message handling functions within the MTP layer 3 differ [29].

The message handling functions consists of three processes : the message routing process, the message distribution process and the message discrimination process. Message routing occurs in both models since this function determines the link over which messages have to be sent and makes load sharing possible. Message distribution occurs at a destination point and determines to which user part (UP) a message is to be delivered. This function is also common to both models.

The message discrimination function occurs only in the quasi - associated model. This process determines if a message has reached its destination point. If a destination point has been reached, the message distribution function is activated. If a message has not reached its destination, the routing function is activated.

3.2 General MTP models

3.2.1 General model for the MTP Level 3

The MTPL3 define transport functions and procedures that are common to the operation of individual signalling links. These functions and procedures fall into two major categories : signalling message handling functions and signalling network management functions.

According to functional specification, the level 3 of the MTP comprises six processes. However, only three processing phases are considered since all control and management functions are omitted. The three processing phases included in the generic submodel are as follows [24]:-

- phase 1 : message routing;
- phase 2 : message discrimination;
- phase 3 : message distribution.

The generic submodel is shown in figure 3.1. It consists of a three phase processor with a separate queue for each processing phase. A message from the UP is delayed by queue 1 and the phase 1 processor before entering the MTPL2. Messages arriving from the MTPL2 would have to first pass through queue 2 before being processed at the phase 2 processor. The phase 2 processor determines if a message has reached a destination point or not. If the point is a destination point the message is sent to the phase 3 processor before entering the UP. If the point has transfer capability, the message is passed back through the phase 1 processor before entering the MTPL1.

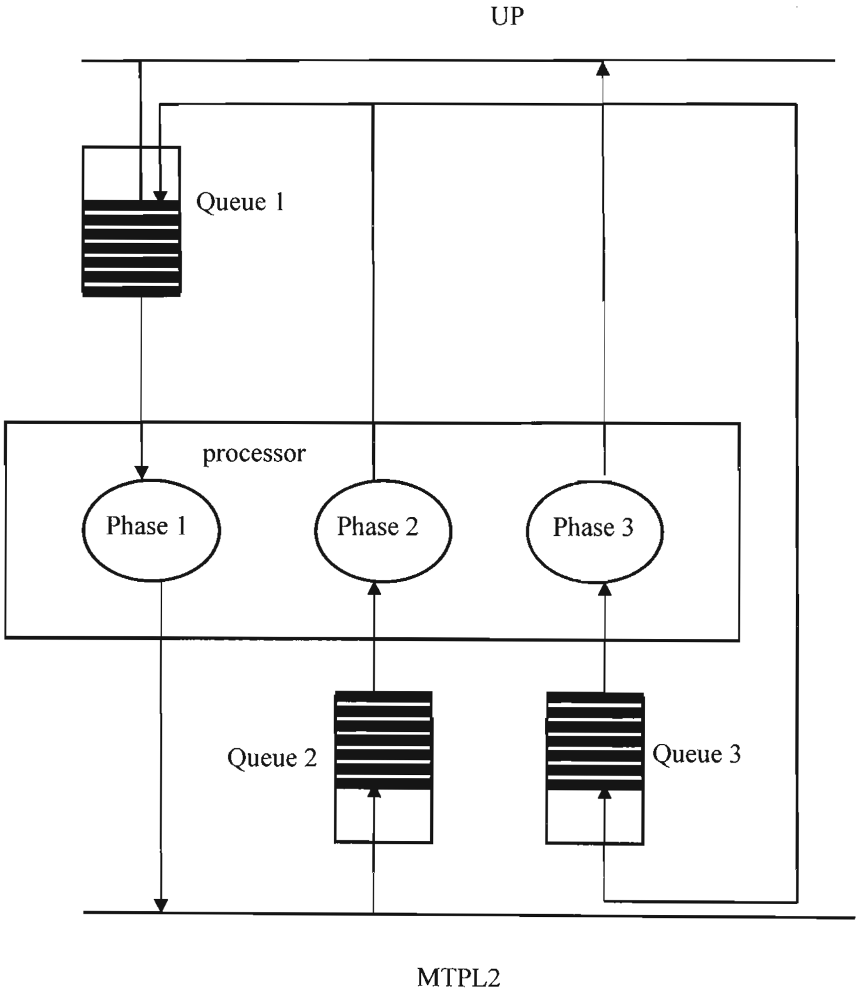


Figure 3.1 : MTP layer 3 model.

3.2.2 Generic model for the MTP layer 2 and MTP layer 1

The MTPL2 is the equivalent of the OSI data link layer and ensures the reliable sequenced delivery of data packets across the MTPL1. The MTPL2 can be modelled by two phases (refer to figure 3.2) :-

- phase 1 : transmission phase and

- phase 2 : reception phase.

These two phases provide the following signalling link functions : signal unit delimitation and alignment, error detection and correction, initial alignment and link error monitoring. A third phase which handles flow control when congestion has been detected at the receiving end of a signalling link has been omitted since this process is used only when the receiving terminal has detected excessive loading and is not in operation for normal loads.

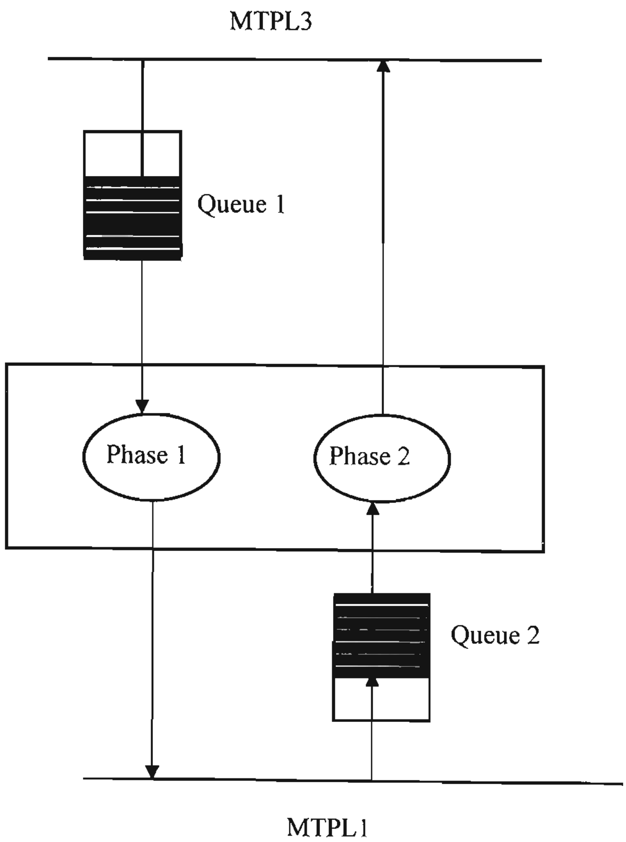


Figure 3.2 : MTP layer 2 model.

The MTPL1 handles the CCS7 signalling data link operations and defines the OSI physical layer characteristics in a CCS7 network. The MTP layer 1 can be modelled as two uni-directional links with propagation delay. The MTP layers form the lower functional blocks of the CCS7 protocol and can be related to the OSI model (refer to chapter 1 : figure 2 for the CCS7 protocol architecture).

3.3 Network Decomposition

The model to be considered will be based on a signalling link interconnecting two signalling nodes. A signalling node can be a signalling point (SP) or a signalling transfer point (STP). The model would have to accommodate for the following relations :-

- link interconnecting SPs;
- link interconnecting a SP and a STP;
- link interconnecting a STP and a SP;
- link interconnecting STPs;

For all cases, the sending/receiving MTP level 2 (MTPL2) and the MTP level 1 (MTPL1) would be similar. Also, a single traffic generator can be used to represent the composite traffic for all UPs while a traffic sink can be used at destination points.

The sending MTPL2 (SL2) consists of the transmission buffer (TB) queue, the processing stage and the emission stages (see figure 3.3). A MSU in the FCFS TB has to first pass through the processing stage before being emitted. Also, a MSU in the processing stage must wait till the emission stage is free before being emitted. The service time in the processing stage is constant for all MSUs whereas the emission times depend on the bit lengths of MSUs and the transmission rates of the data channel. The receiving MTPL2 (RL2) is modelled by a FCFS buffer where MSUs are queued and processed before being sent to the receiving MTPL3.

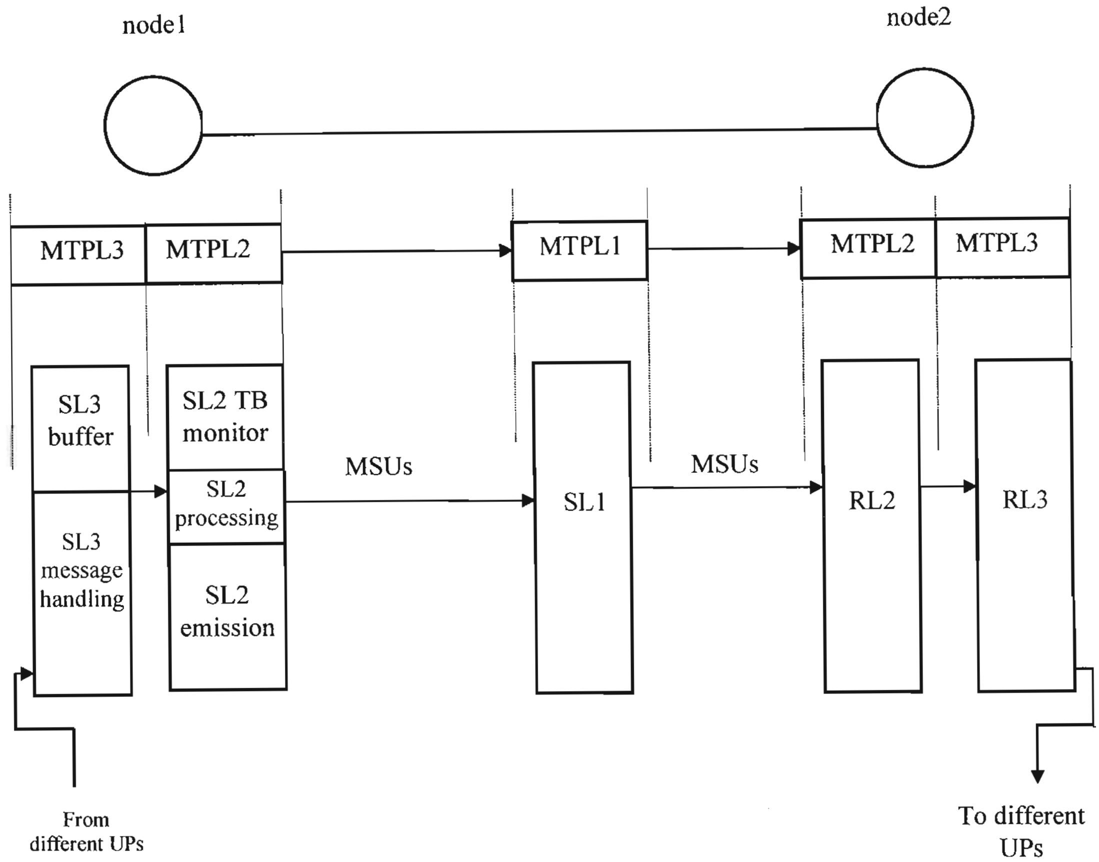


Figure 3.3 : Functional block decomposition of CCS7 elements.

MSUs being emitted into the sending MTPL1 (SL1) would be delayed by the signalling link propagation delay. The signalling link propagation delay is totally dependent on length of links and increases with longer link lengths [39].

The sending/receiving MTPL3 would differ according to the signalling relation since the message handling functions vary for each signalling relation. For each case, the MTPL3 for each signalling node can be modelled as follows (see figure 3.3) :-

- if node1 is a sending SP or STP : the sending MTPL3 can be modelled by a FCFS buffer and a processor for the message handling functions. The message handling function required for sending data is the message routing functions. Message routing is the process of selecting, for each MSU to be sent, the signalling link to be used. Message routing is based on the routing label of the message in combination with the routing data available at the sending signalling point.

  Message routing is destination code dependent with an additional load sharing element that allows traffic to be shared over two or more links. Load sharing may occur over links belonging to the same linkset or links of different linksets.

- if node2 is a receiving SP : The receiving MTPL3 (RL3) is modelled by 2 FCFS queues with processors. The processing time is the delay due to the message discrimination and distribution functions.

  Message discrimination is the process which, upon receipt of a message, determines if the destination signalling node has been reached [26]. This decision is made on the basis of the destination code in the routing label of the MSU. If the discrimination function does not detect a destination SP, then the signalling node has transfer capabilities.

If a destination SP has been detected, then the message distribution function is activated. The distribution function determines by means of the service indicator to which User Part a message is destined.

- if node2 is a STP : the receiving MTPL3 can be modelled by a single FCFS queue with a processor. The processing delay in this case is simply due to the discrimination function which, in this case, determines that node2 is not a destination node but has transfer capability.

For a signalling point - signalling point relation, there is no need for the discrimination functions. A MSU has to traverse a fixed path from origination to destination with no STPs in-between. However, message routing functions would be necessary for load sharing between links belonging to the same link set. Distribution functions is essential for the distribution of MSUs between different UPs.

The processor speed within the MTPL2 coincides with the data rate of links (64kbps). Within the MTPL3, there are no specified processor speeds. This is simply because the processor speeds vary from vendor to vendor. However, the waiting times in queues and the processing delays within the MTPL3 is small in comparison with the MTPL2.

3.4 Implementation of MTP model

The OPNET modeller supports the telecommunication model development process, the generation of specific network models, the management of input and output data, the execution of simulation experiments and the analysis of simulation results. The MTP models described in 3.1 and 3.2 can be implemented in OPNET using a generic layer based model approach [30].

A requirement of this study is to use OPNET to analyse the end to end (ETE) delays experienced in different CCS7 network topologies. This parameter necessitates the implementation of certain MTP functionalities such as that of error allocation/correction, message routing, transmission/propagation delay and the failure/recovery of signalling elements. In this regard, the error and delay characteristics can be implemented in tandem as the MTPL1 and MTPL2 functionalities while the routing and element failure/recovery can be related to the MTPL3.

Figure 3.4 gives the internal configuration for a signalling node that is built up from the different MTP functional blocks. Receivers and transmitters occur between lines 0 and 1 and are labelled as "pt_tx" followed by two numbers for a transmitter; and "pt_rx" followed by numbers for a receiver. These transceivers connect to duplex signalling links from other nodes. The MTPL2 queues occur between lines 1 and 2 (labelled as "q_" followed by a number) while any processes for this layer occurs between lines 2 and 3 (labelled as "p_" followed by a number).

The queues for the MTPL3 occur between lines 3, 4; lines 7, 8; and lines 6, 7 (labelled as "q_" followed by a number). The MTPL3 routing process is represented by the processes between lines 7,8 and 8,9 (labelled as "sdr" and "address"). The UPs are just represented by a "source" and "sink" between lines 8 and 9.

The required MTPL1 and MTPL2 functionalities can be implemented by a point to point pipeline stage. The pipeline stage simply takes care of the necessary MTPL1 and MTPL2 functions. In figure 3.4, the pipeline stage occurs between the receivers and transmitters in the following order : transmission delay stage, closure stage, propagation delay stage, collision stage, error allocation stage and error correction stage. These attributes are defined within the link model stage of OPNET's parameter editor tool.

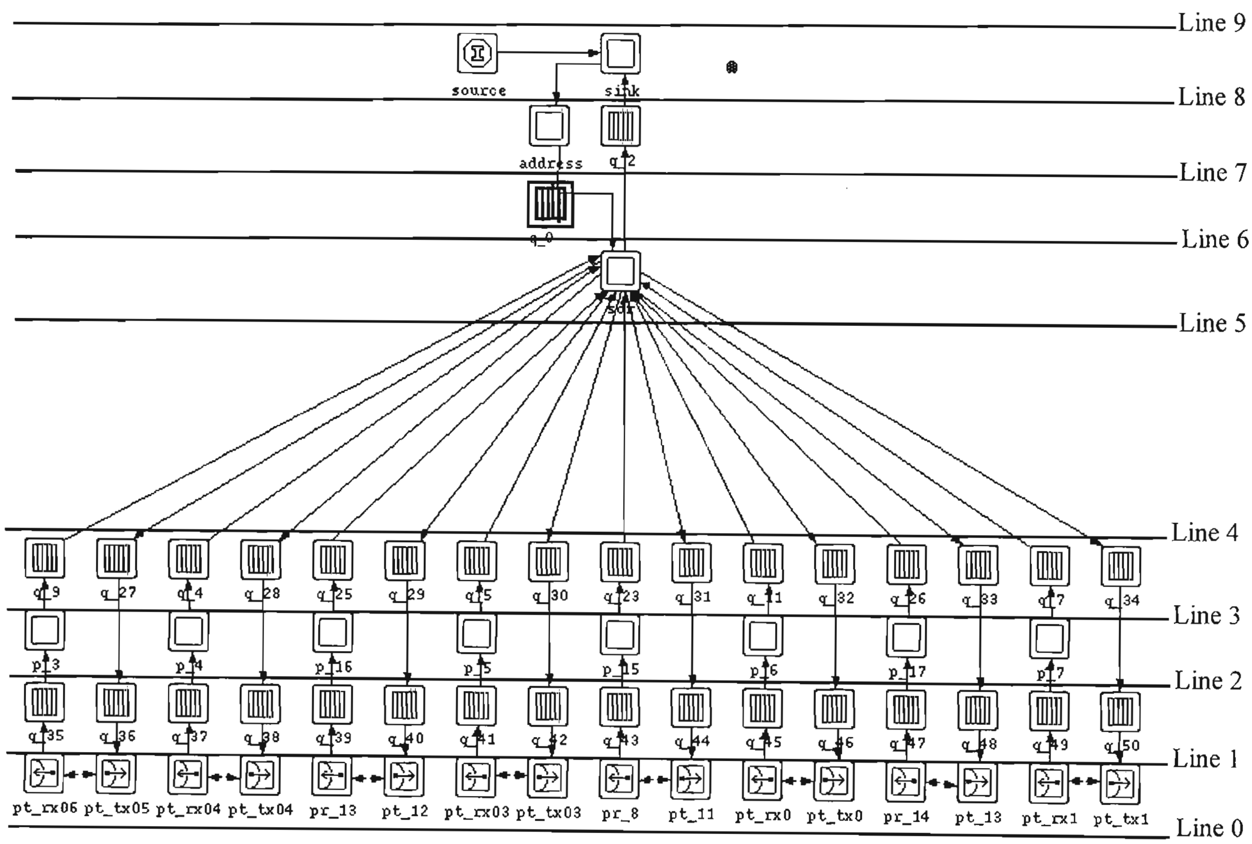


Figure 3.4 : OPNET node model

The transmission delay stage is the first stage of the pipeline, and is specified by the "txdel model" (txdel = transmission delay) of the point to point link [33]. This stage is invoked immediately upon beginning transmission of a packet and is invoked to calculate the amount of time required for the entire packet to complete transmission.

The closure stage is the second stage of the pipeline, and is specified by the "closure model" attribute of the point to point link. This stage occurs immediately after the transmission delay stage with no simulation time elapsing. The purpose of this stage is to determine whether a particular receiver is capable of receiving a transmission.

The propagation delay stage is the third stage of the pipeline, and is specified by the "propdel model" attribute of the point to point link. The purpose of this stage is to calculate the amount of time required for the packet to travel from the transmitter to receiver. The result is generally dependant on factors such as type of physical medium, the distance between source and destination and the frequency of transmissions.

The collision stage occupies the fourth position in the point transceiver pipeline, and is specified by the "coll model" attribute of the point link. This stage is called each time a packet of interest experiences a new collision; a collision is considered to occur when a packet of interest arrives at the receiver while another packet has already begun arriving on the same channel; or when a packet of interest has already begun arriving and another packet enters the channel.

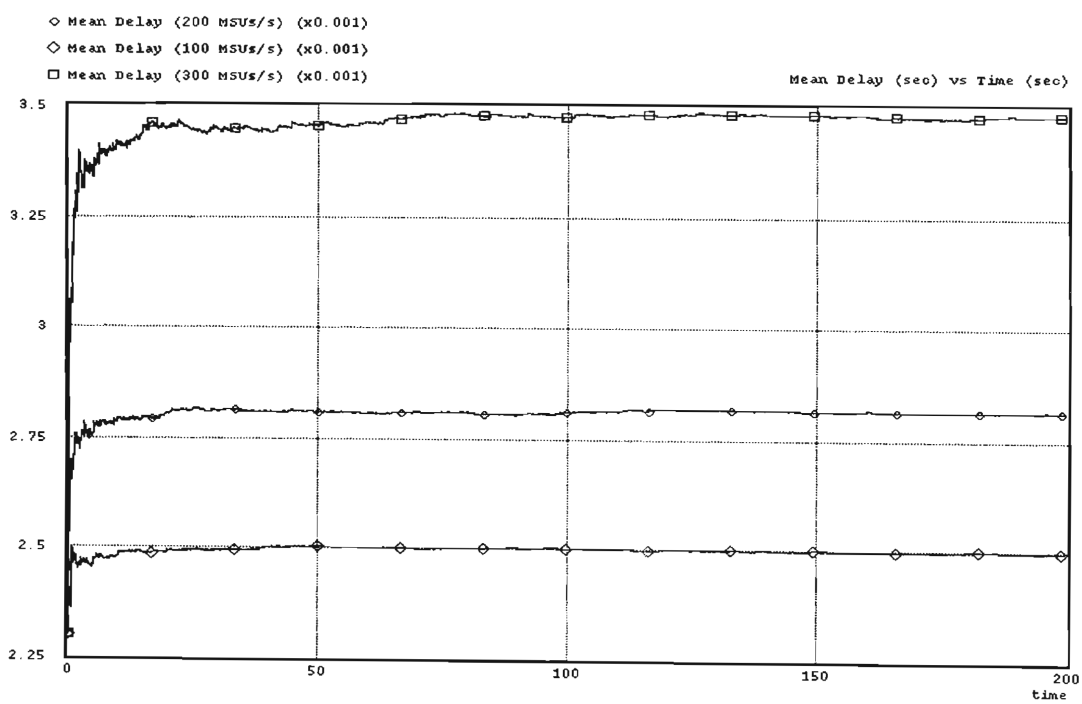
The error allocation stage is the fifth stage of the pipeline, and is specified by the "error model" attribute of the point link. It is invoked after the entire packet has completed reception at the destination, in order to estimate the number of bit errors that occurred.

This is referred to as allocating errors to packets. The result is generally dependent on a bit error probability which reflects the quality of the link, and the packet length.

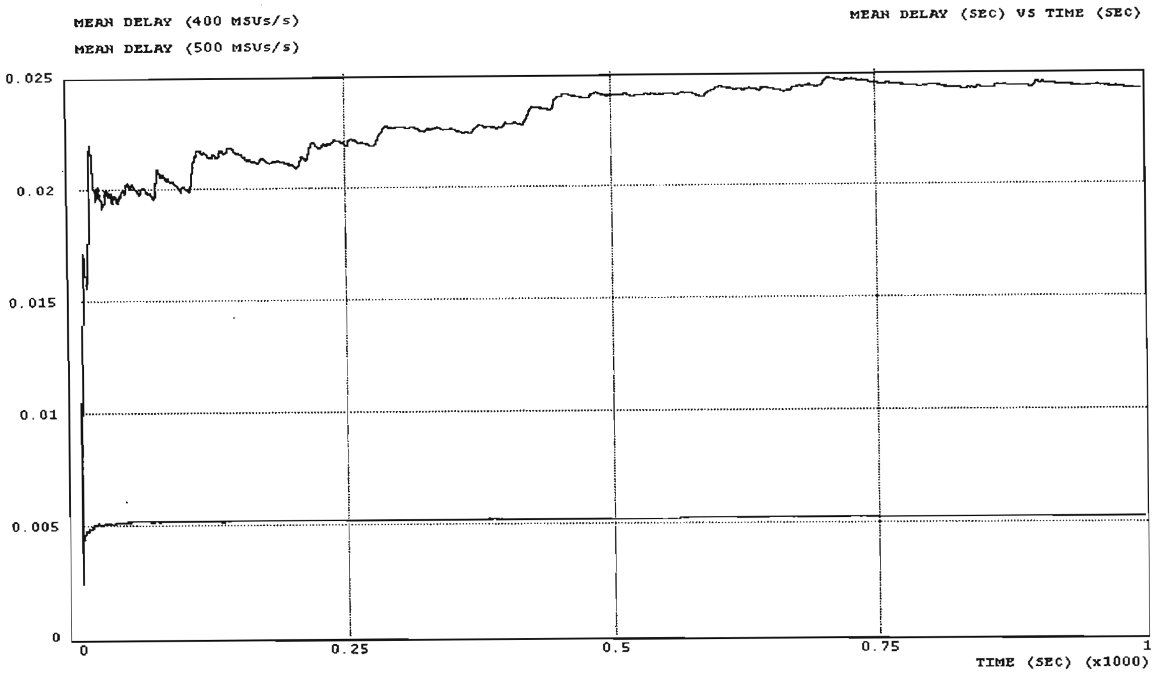
The error correction stage is the sixth and final stage of the pipeline, and is specified by the "ecc model" attribute of the point link. It is invoked immediately after the return of the error allocation stage, with no simulation time elapsing in between. The purpose of this stage is to determine whether or not a packet arriving can be accepted and forwarded to one of the receiving modules in the destination node.

The stage models in the pipeline are defined within OPNET (see table A1 in appendix for the models used). The last stage (ecc model) only forms part of the basic error correction method as defined in CCS7. The basic error correction method requires the negatively acknowledged message and all those transmitted after it to be retransmitted in the order that they were first sent. Simplification of the modelling of the basic error correction protocol can be achieved by associating a 'penalty' to an erroneous MSU arriving at the point to point receiver. The "penalty" is to wait a period of time equal to the signalling loop delay (the signalling loop delay is the time elapsed between the start of emission of the MSU at the sending terminal and the receipt of its negative acknowledgement at the same terminal). This modification to the existing error correction is represented by the process icon between lines 1 and 2 in figure 3.4 (the internal configuration of the icon can be found in the appendix A as figure A2 and the software for the process in A2.1, A2.2, A2.3 and A2.4).

The queue representing the MTPL2 transmission buffer occurs between lines 1 and 2 in figure 3.4. In any implementation of Common Channel Signalling System No. 7 (CCS7), the level 2 transmission buffer is assumed to be a major component of the total queueing delay. It is for this reason that the ITU-T recommendations provides a mathematical model representing the Message Transfer Part (MTP) level 2 buffer only (see section 2.2). Within a SP-SP relation, the MTPL2 transmission buffer can be simulated on OPNET where a zero processing time to all levels (refer to simulations 3.1 and 3.2 – the vertical axis in these simulations represents the mean delay). A zero queue service rate is assigned to all levels except the MTPL2 buffer so that a message would only have to pass through this queue before being emitted onto the data link.



Simulation 3.1 : MTPL2 queueing delays for loadings of 100, 200 and 300 MSUs/s.

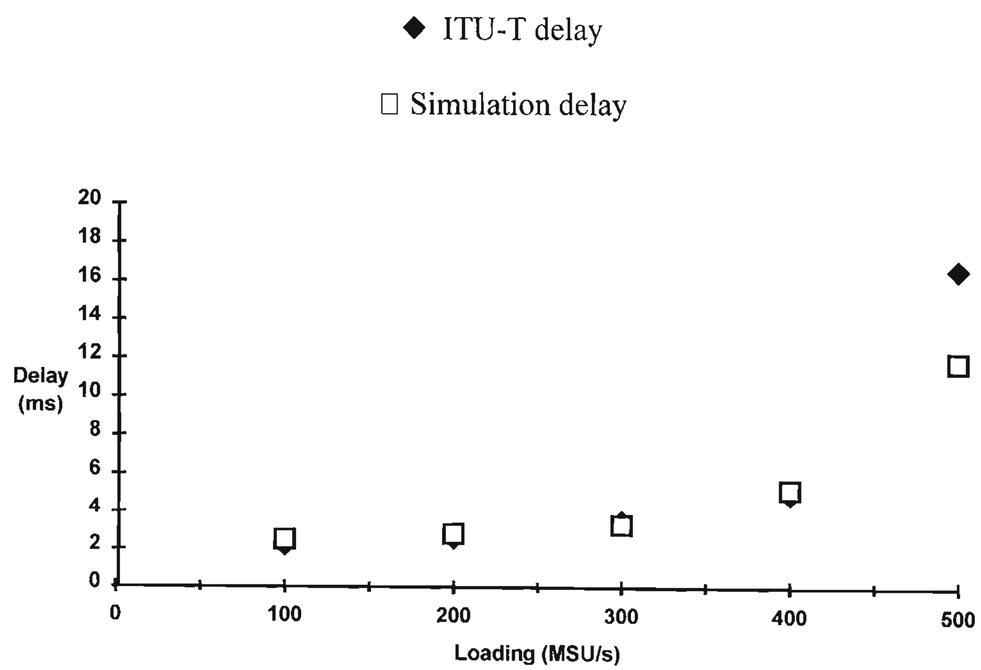


Simulation 3.2 : MTPL2 queueing delays for loadings of 400 and 500 MSUs/s.

The first 10,000 samples obtained from the simulation had to be discarded since these outcomes were affected due to early message arrivals being served too quickly. This is because the TB was empty for the earlier period of the simulation. The OPNET simulations of mean queueing delay for various loading conditions can be compared with computational findings as described by the ITU-T queueing formulae from section 2.2 (refer to table 3.1 and graph 3.1).

Loading (MSUs/s)	Mean delay (s) ITU-T recommendations	Loading (MSUs/s)	Mean delay (s) Simulation
100	0.00247	100	0.0025026
200	0.00284	200	0.0028223
300	0.00351	300	0.0034865
400	0.00518	400	0.0051923
500	0.01656	500	0.0117787

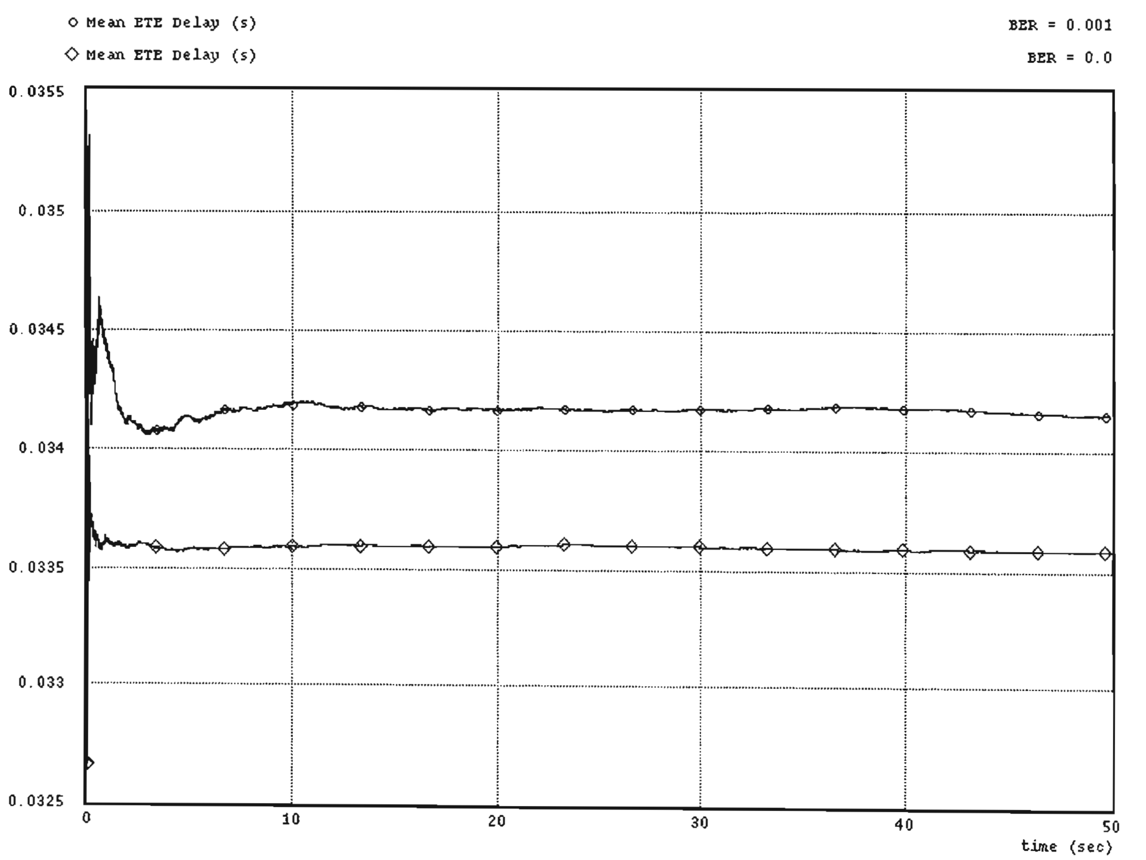
Table 3.1 : Comparison of mean delays



Graph 3.1 : Comparison between mean delays

Simple queueing theory shows that a link does not build up any appreciable MTPL2 queue unless the link is fully utilised (1 Erlang of traffic) [32]. Since the maximum number of MSUs that can be transmitted in 1s = 533 MSUs, the mean queueing delay rises sharply as the loading of MSUs approaches this limit.

The deviation between the analytical and simulated models at high loads is simply due to statistical equilibrium not being reached. This deviation decreases for longer simulations. Even for short simulations, the results do not differ significantly. This simply shows that the simulation model described earlier is consistent with the ITU-T defined MTPL2 queueing formulae.



Simulation 3.3 : End to end delays with errors.

When errors occur, the modified error correction protocol defined within OPNET increases the end to end delay of packets. This can be seen in simulation 3.3 where the end to end delay for a bit error rate (BER) of 0.001 is greater than that for a BER of 0. This increase in simulation time is due to the signalling loop delay being added to messages being in error.

A function of the MTPL3 is routing, choosing the path along which packets travel from their source to their destination. The MTPL3 sends packets through networks by means of the routing, discrimination and distribution functions. The queues for these functions occur between lines 7 and 8 and also lines 3 and 4 in figure 3.4. The OPNET defined Bellman-Ford shortest path algorithm will be used to replace these three MTPL3 message handling functions (refer to the process icon between lines 5 and 6 in figure 3.4). The internal configuration of this icon can be found in figure A3 in appendix A3 while the modified header block, functional block, temporary variable block and init programme for the routing process can be found in A3.1 to A3.4 (appendix A3). All other programmes for the routing process are standard to OPNET.

With shortest path routing, networks can be represented by a weighted, directed graph, with nodes represented as vertices and links as edges. The weight of an edge corresponds to the cost of sending data across the link. The cost may take into account many parameters, typically including capacity and delay of the link. The best route from one node to another is the path with the lowest total cost.

The routing process is made possible by means of the user defining the point code and the destination code for each node. A user defined cost value is also required for each signalling link where the lowest cost represents the best route. An example of the routing protocol choosing paths for user defined cost values can be seen in figures 3.5 and 3.6. In figure 3.5, the shortest path follows route 1 : SPA-STPB-STPC-SPF (lowest cost calculation = 0.1 + 0.1 + 0.1 = 0.3); while the shortest path in figure 3.6 follows route 2 : SPA-STPD-STPE-STPC-SPF (lowest cost calculation = 0.1 + 0.1 + 0.1 + 0.1 = 0.4). A different number of nodes and links are traversed by messages from source to destination from route 1 to route 2 and there is an obvious difference between the end to end delays for these paths. The result can be seen in simulation 3.4 where the lower simulation represents route 1 while the upper simulation refers to route 2. In simulation 3.4, the vertical axis represents the mean delay in seconds.

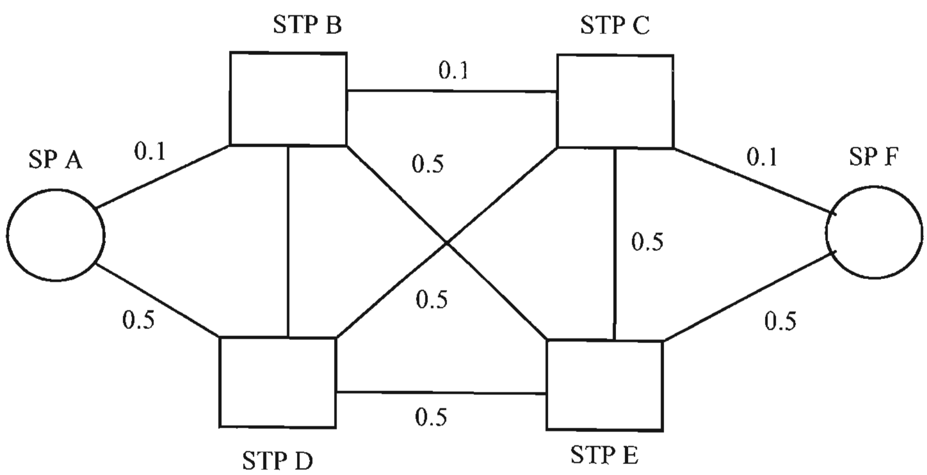


Figure 3.5 : Cost calculations in quad structure

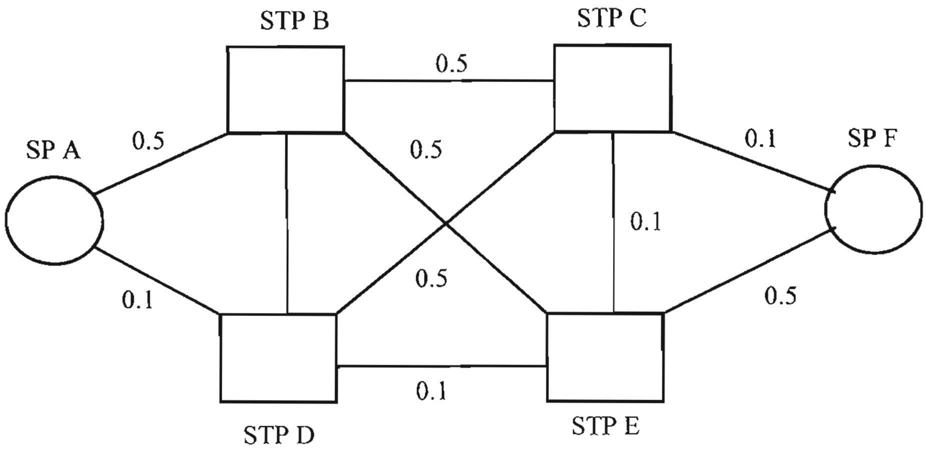
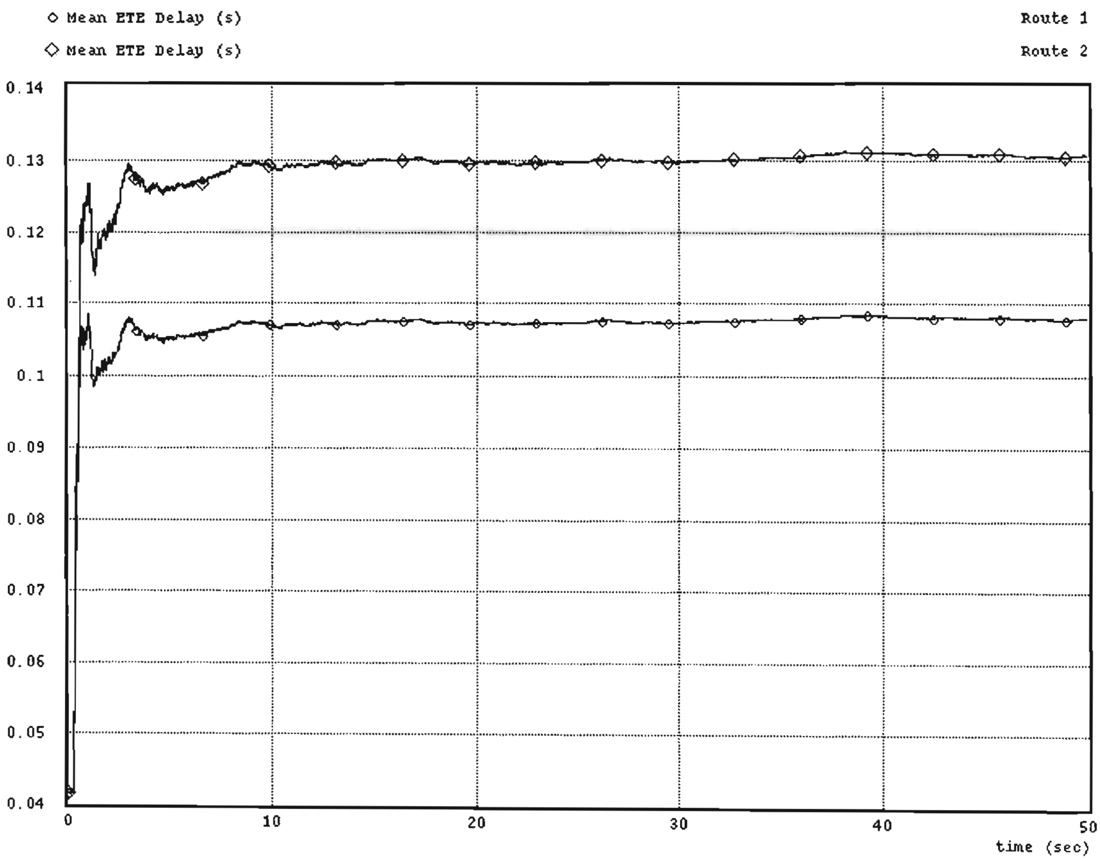


Figure 3.6 : Cost calculations in quad structure.



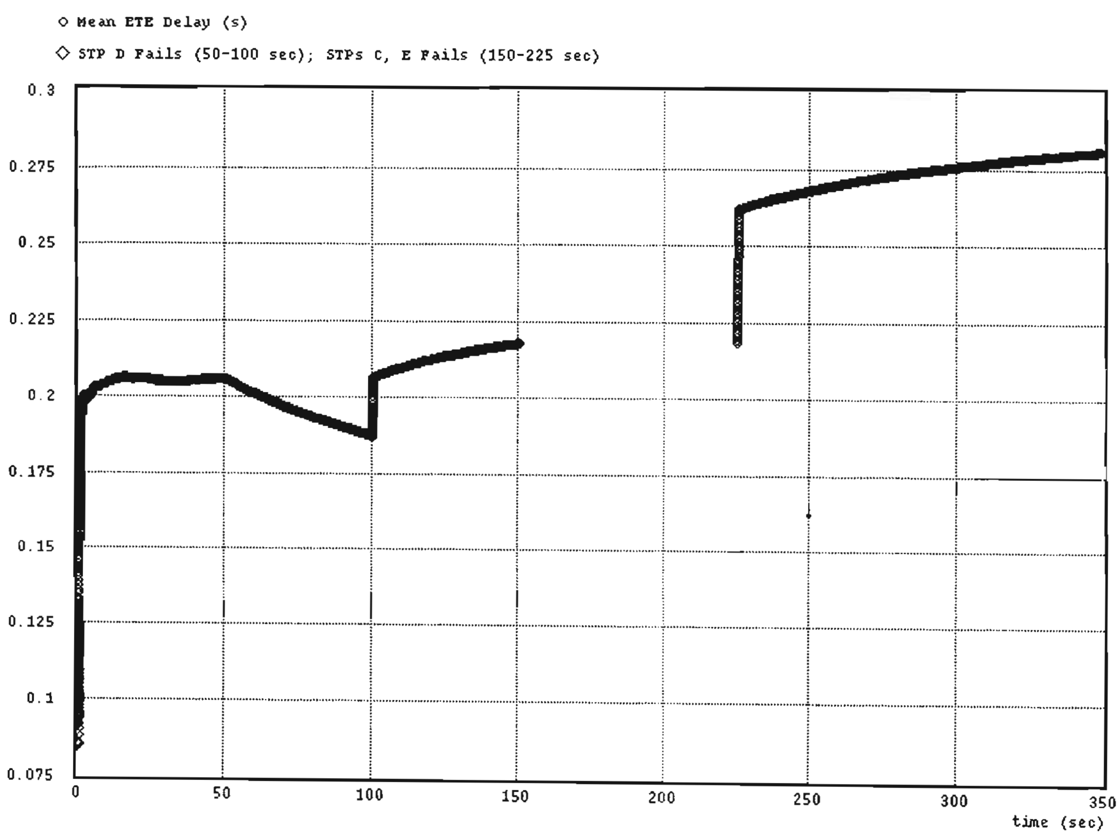
Simulation 3.4 : End to end delays for two different routes.

At the initialisation of the routing process, each node constructs an interface table, which contains information about each node network interfaces, including the interface type (direct point to point links), cost and interface address (node address). The node address is usually self assigned by the software but can be modified to be user-assigned (refer to A3.4 in appendix A3). This user assigned node address is then assigned to packets by the "address" process icon between lines 7 and 8 in figure 3.4 (refer to figure A4, A4.1 and A4.2 in appendix A4 for the process icon's internal configuration).

The node then broadcasts an address resolution packet (ADP) across each of these interfaces, informing all nodes on each local network (point to point link) of its interface address on that local network and its node address.

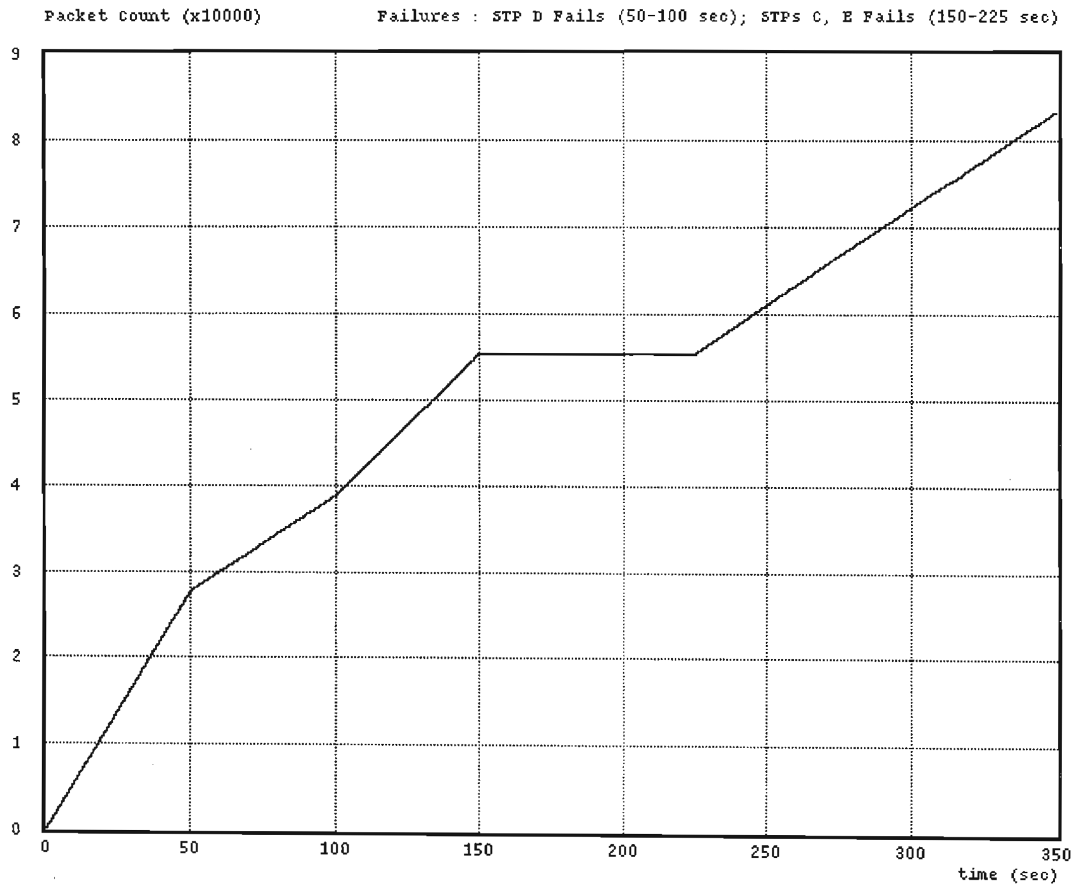
When a node receives an ADP, it considers this specified route for inclusion in the routing table. If the route reaches a previously unreachable node, or if it is less expensive than the previously known route, it is added to the routing table, and a route advertisement packet (RAD) describing the new route is constructed and broadcast through all network interfaces. A node receiving a RAD compares the advertised route with the routes in the nodes routing table. If this new route is less costly than the best known route to the advertised destination, or if it leads to a previously unreachable node, this new route is added to the routing table.

The failure and recovery of a node may be implemented in the routing process. If the routing process is stopped then that node becomes non-functional. The modification of the routing process to allow failure and recovery of nodes can be seen in appendix A3 (A3.1 - A3.4). In figure 3.6, STP D was failed at 50s and recovered at 100s while STPs C and E was failed at 150s and recovered at 225s. The failure and recovery of these nodes can be seen in simulation 3.5. The end to end delay clearly shows the failure periods and increases with errors. The “space” during the simulation simply indicates that there is a end-to-end failure (in this case mated STPs C and E fail). This result may be also visualised in simulation 3.6 where the packet count decreases for single nodal failure and becomes constant for end-to-end failure.



Simulation 3.5 : End to end delays with node failures

The user parts (UPs) may be visualised as a source and sink for messages. The source is simply an ideal generator termed "source" between lines 8 and 9 in figure 3.4. The destroying of packets is handled by the "sink" process icon between lines 8 and 9 in figure 3.4 (refer to A5.1 - A5.6 and figure A5 in appendix A5 for internal configuration of process icon).



Simulation 3.6 : Packet count with node failures

The overall model implementation within OPNET outlines the necessary functions required to send packets through the CCS7 network. The simulations of the MTPL2 queueing delays (simulations 3.1 and 3.2) prove that the model is consistent with the ITU-T queueing formulae for that layer. Simulations 3.3-3.6 exhibits the routing protocol applied to a network under failure conditions, with errors and for different path lengths.

Thus, simulations of the end-to-end delays associated with different networks can be obtained. Further, simulations can also be obtained as network elements fail and recover. This can be used to test the delay performance characteristics of any network.

4. Network Topologies Analysis

4.1 Introduction

There are two basic signalling modes : the associated and quasi-associated modes of signalling. Each signalling mode presents both advantages and drawbacks according to the criteria used to set up the signalling network; these criteria are numerous and may be as different as minimal cost, reliability, geographical constraints and/or services planned [3], [34].

In the associated mode of signalling, the messages relating to a signalling relation between two signalling points are conveyed over a signalling link set which directly interconnects these two points. When considering the telephony aspects, it can be said that, in the associated mode, the telephone network and the signalling network are parallel : the two exchanges involved are both interconnected by one or several signalling links.

Taking into account the capabilities of a digital signalling link, which is able to control the signalling traffic of several hundred telephony circuits, the associated mode seems well adapted to serve large speech circuit groups. The associated mode allows network designers to take advantage of the direct interconnection in terms of transmission medium and also to minimize the signalling end to end delay.

In the quasi-associated mode of signalling, the messages relating to a signalling relation are conveyed over two or more signalling link sets in tandem passing through

STPs, the path of each message being predetermined in each node of the signalling network. The quasi-associated structure allows, in conjunction with a signalling routing policy, the achievement of the concept of total accessibility between the SPs, without making mandatory the total meshing of nodes. The total accessibility is defined as the possibility for a message, in a given CCS7 network, to be routed from any originating point to any destination point using only the MTP function.

This section will concentrate on topologies common to CCS7 : mesh, hierarchical, quad and modifications to these structures. The availability and reliability of these CCS7 networks would be evaluated utilising a simplified approach [10], while an approximation of the end to end delays (ETE) can be obtained by simulation. Utilising these performance parameters, a suitable topology can be selected to transform an existing signalling network to one that can meet the demands of present telecommunication services.

The availability and reliability analysis will be centred on works by G. Steyn and H. Hanrahan [10], and utilises the mean time to restore (MTTR) and the mean time between failures (MTBF). The values for the MTTR and MTBF can be obtained from table 4.1. The availability and reliability for all signalling links are assumed to have negligible effect on the calculations of availability and reliability. The definitions for the reliability and availability were given in chapter 2.

Network Element Type	MTBF	MTTR
SEP (signalling end point)	525 597 (minutes/year)	3 (minutes/year)
STP (signalling transfer point)	524 574 (minutes/year)	26 (minutes/year)

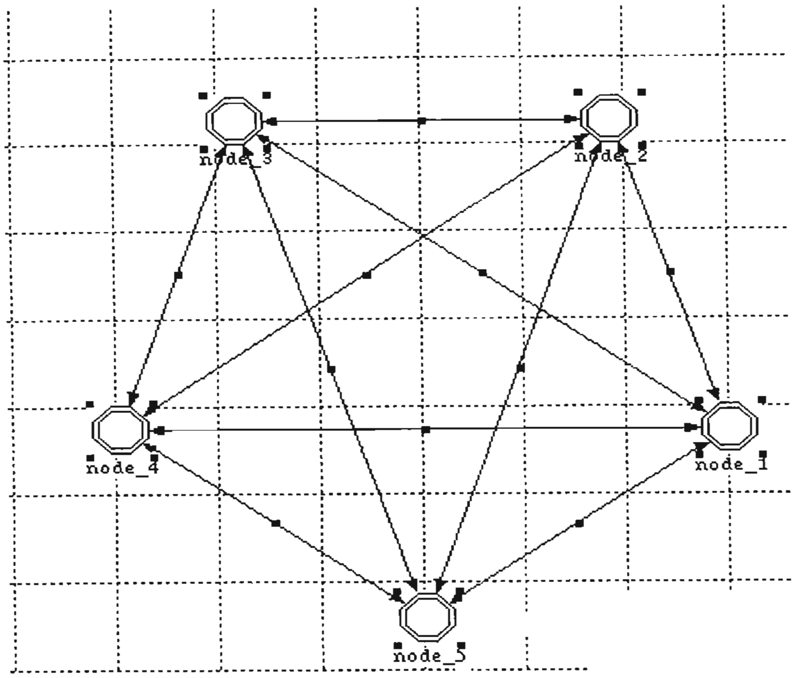
Table 4.1 : Values for MTTR and MTBF for CCS7 network nodes

4.2 Mesh topology

In this signalling network structure, the associated mode of signalling is generally used. This signalling network seems only viable in the case of a network composed of a small number of signalling points. In a fully meshed network structure composed of N signalling points, the number of signalling linksets needed is :

$$\text{No. signalling linksets} = \frac{N(N-1)}{2} \qquad 4.1$$

The mesh topology has 100% redundancy where any failure causes traffic to be diverted to alternative paths. Each signalling component within this topology should be engineered to allow twice its peak load. For an availability and reliability analysis, a five node topology was chosen, refer to figure 4.1, where the routes defined traversed the four shortest paths, refer to table 4.1. In this section, only four paths were defined for all networks in order to simplify calculations. The origination node for the specified route was taken to be node 1 (N1) while the destination point was node 2 (N2).



*all nodes are SPs

Figure 4.1 : Five node mesh topology

Route Name	Path Followed (NX - node X; XLY - link connecting nodes X and Y; X,Y – integers)				
T	N4	3L4	N3	2L3	N2
V	N4	4L2	N2		
W	N4	4L1	N1	1L2	N2
X	N4	4L5	N5	5L1	N2

Table 4.2 : Defined routes for mesh topology consisting of SPs

The expression for the availability for the routes in table 4.1 is given by equation 4.2.1 :

$$
\begin{aligned}
P[T \cup V \cup W \cup X] = {} & P[T] + P[V] + P[W] + P[X] \\
& - P[T \cap V] - P[T \cap W] - P[T \cap X] \\
& - P[V \cap W] - P[W \cap X] - P[V \cap X] \\
& + P[T \cap V \cap W] + P[T \cap V \cap X] \\
& + P[T \cap W \cap X] + P[V \cap W \cap X] \\
& - P[T \cap V \cap W \cap X]
\end{aligned}
\qquad 4.2.1
$$

The terms for the expressions defined in equation 4.2.1 can be given by equation 4.2.2 :

$$
\begin{aligned}
P[T] &= A_T \\
P[V] &= A_V \\
P[W] &= A_W \\
P[X] &= A_X
\end{aligned}
$$

$$
\begin{aligned}
P[T \cap V] &= P[T]P[V/T] = A_{4L2} . A_T \\
P[T \cap W] &= P[T]P[W/T] = A_T A_{4L1} A_{N1} A_{1L2} \\
P[T \cap X] &= P[T]P[X/T] = A_T A_{4L5} A_{N5} A_{5L1} \\
P[V \cap W] &= P[V]P[W/V] = A_V A_{4L1} A_{N1} A_{1L2} \\
P[W \cap X] &= P[W]P[X/W] = A_W A_{4L5} A_{N5} A_{5L1} \\
P[V \cap X] &= P[V]P[X/V] = A_V A_{4L5} A_{N5} A_{5L1}
\end{aligned}
$$

$$
\begin{aligned}
P[T \cap V \cap W] &= P[T]P[V/T]P[W/V,T] \\
&= A_T A_{4L2} A_{4L1} A_{N1} A_{1L2} \\
P[T \cap V \cap X] &= P[T]P[V,T]P[X/V,T] \\
&= A_T A_{4L2} A_{4L5} A_{N5} A_{5L1} \\
P[T \cap W \cap X] &= P[T]P[W/T]P[X/W,T] \\
&= A_T A_{4L1} A_{N1} A_{1L2} A_{4L5} A_{N5} A_{5L1} \\
P[V \cap W \cap X] &= P[V]P[W/V]P[X/W,V] \\
&= A_V A_{4L1} A_{N1} A_{1L2} A_{4L5} A_{N5} A_{5L1}
\end{aligned}
$$

$$
\begin{aligned}
P[T \cap V \cap W \cap X] &= P[T]PV/T]P[W/V,T]P[X/W,V,T] \\
&= A_T A_{4L2} A_{4L1} A_{N1} A_{1L2} A_{4L5} A_{N5} A_{5L1}
\end{aligned}
\qquad 4.2.2
$$

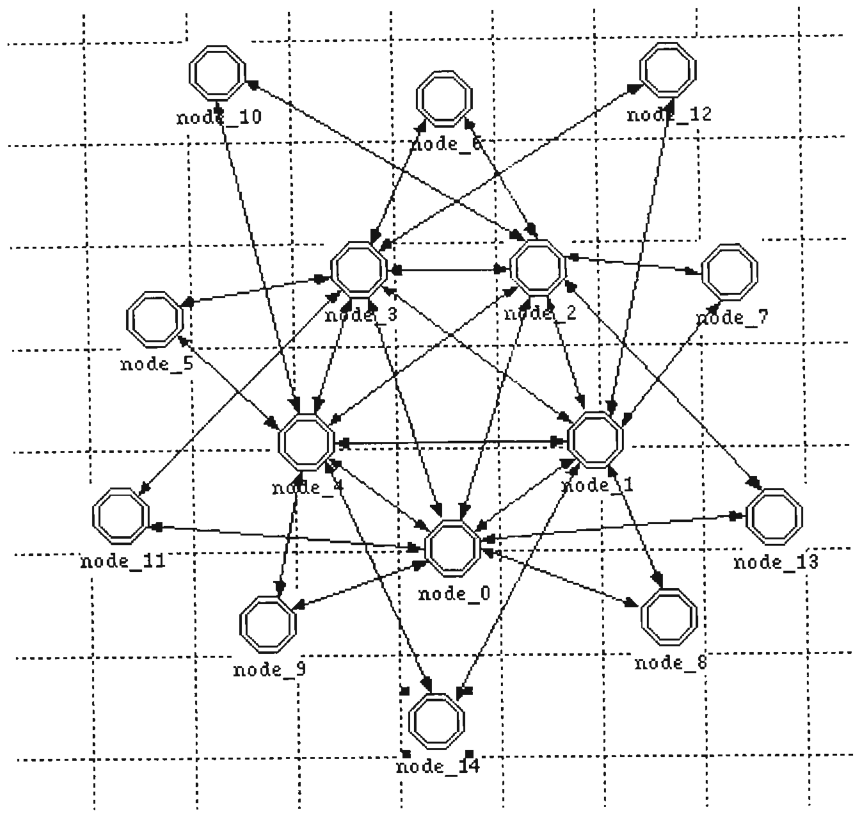
The reliability for these routes can be given by equation 4.2.3 and can be obtained in the same way as for the availability :

$$\begin{aligned} R(t) = {} & R_T + R_V + R_W + R_X - R_T \\ & - R_{N1}R_T - R_{N5}R_T - R_{N1}R_V \\ & - R_{N5}R_V - R_W R_{N5} + R_T R_{N1} \\ & + R_T R_{N5} + R_T R_{N5} R_{N1} + R_V R_{N5} R_{N1} \\ & - R_T R_{N5} R_{N1} \end{aligned} \qquad 4.2.3$$

The availability and reliability can be calculated using Mathlab, where the MTTR and MTBF values were obtained from table 4.1. The following values were obtained for the mesh topology :-

- Availability - 525 596. 999 931 508 1 minutes/year
- Unavailability - 3. 000 068 491 899 82 minutes/year
- Reliability - 0. 999 532 167 255 86

The generalised mesh network is a backbone network of fully connected STPs where clusters of SPs are connected to different pairs of STPs (refer to figure 4.2). In this case, if a STP fails, its load is split to a number of STPs. Signalling networks built on STP functions could be considered as having hierarchical structures in which SPs constitute the lower level while STPs represent the higher level. An example of this structure is the quad topology in section 4.3.

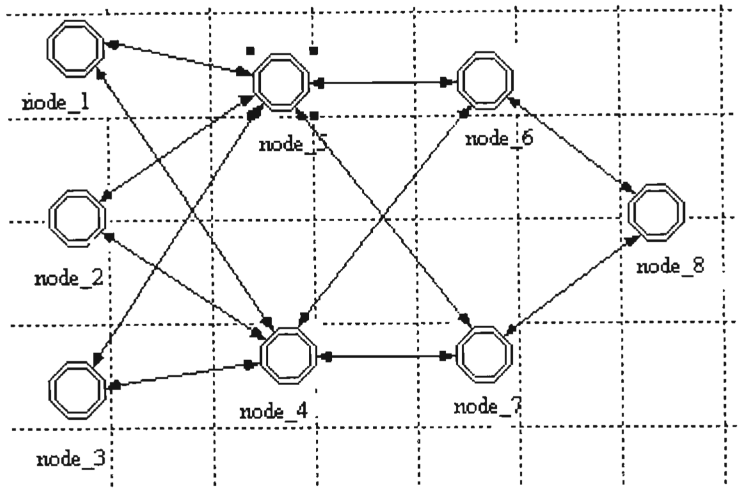


*nodes 0-4 are STPs; all other nodes are SPs

Figure 4.2 : Generalised mesh topology with STPs

4.3 Quad topology

The standard quad structure consists of four fully meshed STPs that are "mated" on a pairwise basis (refer to figure 4.3). The quad network has 100% redundancy : that is, for any STP failure, the traffic can be diverted to alternate paths that do not increase the number of transfer points [35]. Therefore, networks must be engineered in such a way that each component can handle twice its peak load under normal conditions. The four shortest routes used for an availability and reliability analysis are defined in table 4.2 where the source and destination nodes are assumed to be 2 and 8 for a route set. This is the topology that has been a success and is currently being deployed in North America.



Nodes 4-7 are STPs; all other nodes are SPs

Figure 4.3 : Standard quad topology

Route Name	Path Followed (NX - node X; XLY - link connecting nodes X and Y; X,Y – integers)						
T	N2	2L4	N4	4L7	N7	7L8	N8
V	N2	2L4	N4	4L6	N6	6L8	N8
W	N2	2L5	N5	5L6	N6	6L8	N8
X	N2	2L5	N5	5L7	N7	7L8	N8

Table 4.3 : Defined routes for quad topology

The expression for the availability of routes can be given in equation 4.3.1.

$$P[T \cup V \cup W \cup X] = P[T] + P[V] + P[W] + P[X] - P[T \cap V] - P[T \cap W] - P[T \cap X] - P[V \cap W] - P[W \cap X] - P[V \cap X] + P[T \cap V \cap W] + P[T \cap V \cap X] + P[T \cap W \cap X] + P[V \cap W \cap X] - P[T \cap V \cap W \cap X] \quad 4.3.1$$

The terms for the expressions defined in 4.3.1 can be given in equations 4.3.2.

$$P[T] = A_T$$
$$P[V] = A_V$$
$$P[W] = A_W$$
$$P[X] = A_X$$

$$P[T \cap V] = P[T]P[V/T] = A_{N6} \cdot A_T A_{4L6} \cdot A_{6L8}$$
$$P[T \cap W] = P[T]P[W/T] = A_T A_{2L5} A_{N5} A_{5L6} A_{N6} \cdot A_{6L8}$$
$$P[T \cap X] = P[T]P[X/T] = A_T A_{2L5} A_{N5} A_{5L7}$$
$$P[V \cap W] = P[V]P[W/V] = A_V A_{2L5} A_{N5} A_{5L6}$$
$$P[V \cap X] = P[W]P[X/W] = A_V A_{2L5} A_{N5} A_{5L7} A_{N7} \cdot A_{7L8}$$
$$P[W \cap X] = P[V]P[X/V] = A_W A_{5L7} A_{N5} A_{5L7} A_{N7} \cdot A_{7L8}$$

$$P[T \cap V \cap W] = P[T]P[V/T]P[W/V,T] = A_T A_{4L6} A_{N6} \cdot A_{6L8} A_{2L5} A_{N5} A_{2L5}$$
$$P[T \cap V \cap X] = P[T]P[V,T]P[X/V,T] = A_T A_{4L6} A_{N6} A_{6L8} A_{2L5} A_{N5} A_{5L7}$$
$$P[T \cap W \cap X] = P[T]P[W/T]P[X/W,T] = A_T A_{2L5} A_{N5} A_{5L6} A_{6L8} A_{N6} A_{5L7}$$
$$P[V \cap W \cap X] = P[V]P[W/V]P[X/W,V] = A_V A_{2L5} A_{N5} A_{5L6} A_{5L7} A_{N7} A_{7L8}$$

$$P[T \cap V \cap W \cap X] = P[T]PV/T]P[W/V,T]P[X/W,V,T] = A_T A_{4L6} A_{6L8} A_{N6} A_{2L5} A_{5L6} A_{N5} A_{5L7} \quad 4.3.2$$

The reliability for the quad topology can be given by equation 4.3.3 and can be obtained in the same way as the availability equations.

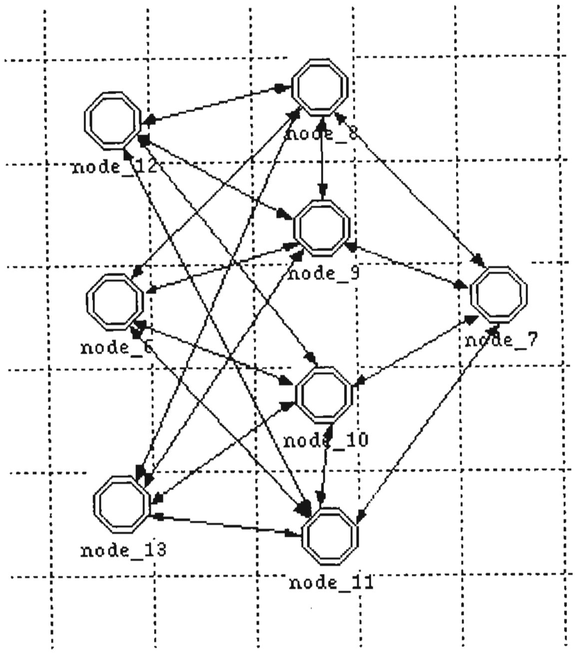
$$
\begin{aligned}
R(t) = {} & R_T + R_V + R_W + R_X \\
& - R_T R_{N6} - R_{N5} R_{N6} R_T - R_{N5} R_T \\
& - R_{N5} R_V - R_{N5} R_V R_{N7} - R_W R_{N7} \\
& + R_T R_{N6} R_{N5} + R_T R_{N6} R_{N5} \\
& + R_T R_{N5} R_{N6} + R_V R_{N5} R_{N7} \\
& - R_T R_{N6} R_{N5}
\end{aligned}
\qquad 4.3.3
$$

The availability and reliability can be calculated using Mathlab, where the MTTR and MTBF values were obtained from table 4.1. The following values were obtained for the quad topology :-

- Availability - 525 593. 975 664 28 minutes/year
- Unavailability - 6. 024 335 218 630 minutes/year
- Reliability - 0. 996 194 834 655 36

4.4 Telstra's new CCS7 topology

Telstra's CCS7 topology was introduced to replace the standard quad structure and is currently being deployed in Australia. The structure has the same number of signalling elements as the quad structure but in this case each SP is connected to four STPs (nodes 12, 6, 7 and 13 are SPs while nodes 8, 9, 10 and 11 are STPs in figure 4.4). In this network the STP planes are not connected since there are different switching technologies. Table 4.3 defines the four shortest paths between nodes 6 and 7 used in an availability and reliability analysis.



*nodes 8-11 are STPs; all other nodes are SPs

Figure 4.4 : New CCS7 topology

The defined routes for the new topology can be found in table 4.4.

Route Name	Path Followed (NX - node X; XLY - link connecting nodes X and Y; X,Y – integers)				
T	N6	6L8	N8	8L7	N7
V	N6	6L9	N9	9L7	N7
W	N6	6L10	N10	10L7	N7
X	N6	6L11	N11	11L7	N7

Table 4.4 : Defined routes for the new topology

The expression for availability is defined in equation 4.4.1.

$$
\begin{aligned}
P[T \cup V \cup W \cup X] = {} & P[T] + P[V] + P[W] + P[X] \\
& - P[T \cap V] - P[T \cap W] - P[T \cap X] \\
& - P[V \cap W] - P[W \cap X] - P[V \cap X] \\
& + P[T \cap V \cap W] + P[T \cap V \cap X] \\
& + P[T \cap W \cap X] + P[V \cap W \cap X] \\
& - P[T \cap V \cap W \cap X] \qquad 4.4.1
\end{aligned}
$$

The terms for the expressions in equations 4.4.1 is given in equation 4.4.2.

$$
\begin{aligned}
P[T] &= A_T \\
P[V] &= A_V \\
P[W] &= A_W \\
P[X] &= A_X
\end{aligned}
$$

$$
\begin{aligned}
P[T \cap V] &= P[T]P[V/T] = A_{N9} . A_T A_{6L9} . A_{9L7} \\
P[T \cap W] &= P[T]P[W/T] = A_T A_{6L10} A_{N10} A_{10L7} \\
P[T \cap X] &= P[T]P[X/T] = A_T A_{6L11} A_{N11} A_{11L7} \\
P[V \cap W] &= P[V]P[W/V] = A_V A_{2L5} A_{N5} A_{5L6} \\
P[V \cap X] &= P[W]P[X/W] = A_V A_{6L11} A_{N11} A_{11L7} \\
P[W \cap X] &= P[V]P[X/V] = A_W A_{6L11} A_{N11} A_{11L7}
\end{aligned}
$$

$$
\begin{aligned}
P[T \cap V \cap W] &= P[T]P[V/T]P[W/V,T] \\
&= A_T A_{6L9} A_{N9} . A_{9L7} A_{6L10} A_{N10} A_{10L7} \\
P[T \cap V \cap X] &= P[T]P[V,T]P[X/V,T] \\
&= A_T A_{6L9} A_{N9} A_{9L7} A_{6L11} A_{N11} A_{11L7} \\
P[T \cap W \cap X] &= P[T]P[W/T]P[X/W,T] \\
&= A_T A_{6L10} A_{N10} A_{10L7} A_{6L11} A_{N11} A_{11L7} \\
P[V \cap W \cap X] &= P[V]P[W/V]P[X/W,V] \\
&= A_V A_{6L10} A_{N10} A_{10L7} A_{6L11} A_{N11} A_{11L7}
\end{aligned}
$$

$$
\begin{aligned}
P[T \cap V \cap W \cap X] &= P[T]PV/T]P[W/V,T]P[X/W,V,T] \\
&= A_T A_{6L9} A_{9L7} A_{N9} A_{6L11} A_{11L7} \\
&\quad A_{N11} A_{11L7} A_{6L11} A_{N11} A_{11L7} \qquad 4.4.2
\end{aligned}
$$

The reliability can be given by equation 4.4.3 and may be obtained in the same way as the availability.

$$R(t) = R_T + R_V + R_W + R_X - R_T R_{N9} - R_{N10} R_T - R_{N11} R_T - R_{N10} R_V - R_{N11} R_V$$
$$- R_W R_{N11} + R_T R_{N9} R_{N10} + R_T R_{N10} R_{N11} + R_T R_{N10} R_{N11} + R_V R_{N10} R_{N11}$$
$$- R_T R_{N9} R_{N11} R_{N11} \qquad 4.4.3$$

The availability and reliability can be calculated using Mathlab, where the MTTR and MTBF values were obtained from table 4.1. The following values were obtained for the new topology :-

- Availability - 525 594. 000 017 123 3 minutes/year
- Unavailability - 5. 999 982 876 760 51 minutes/year
- Reliability - 0. 996 202 033 816 79

4.5 Hierarchical topology

The hierarchical structure evolved from the tree structure with the realization of node disjoint routes between two nodes in a network [36]. The capacity of this structure can be classified by the number of levels where the highest level corresponds to STPs [37]. In figure 4.5, the highest level is represented by nodes 17 and 18 while the next lower level represents the intermediate SPs (nodes 13-16). The bottom layer in this case represents the origination and destination signalling points (nodes 11, 12 and 19). These networks can be evolved into higher level networks but this has an affect of increasing signalling delays. Table 4.4 defines the four shortest paths between nodes 1 and 12 used for an availability and reliability analysis.

The defined routes for the hierarchical topology can be found in table 4.5.

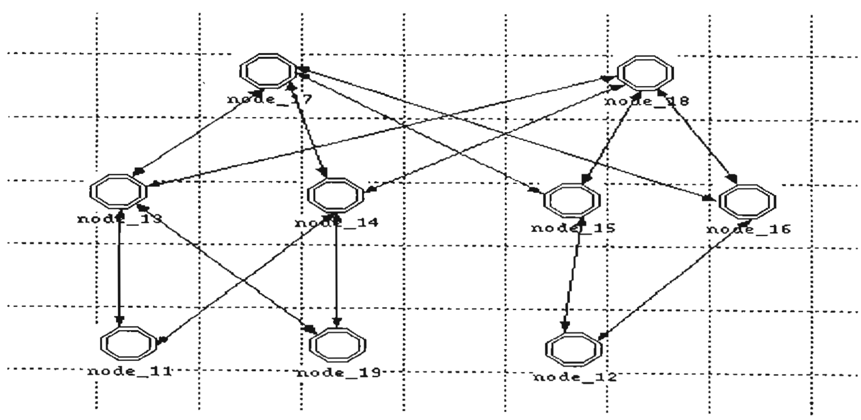
Route Name	Path Followed (NX - node X; XLY - link connecting nodes X and Y; X,Y – integers)								
T	N11	N13	13L17	N17	17L15	N15	15L12	11L13	N12
V	N11	11L14	N14	14L17	N17	17L16	N16	16L12	N12
W	N11	11L13	N13	13L18	N18	18L15	N15	15L12	N12
X	N11	11L14	N14	14L18	N18	18L16	N16	16L12	N12

Table 4.5 : Defined routes for hierarchical topology.

The availability expression for these routes can be found in equation 4.5.1.

$$\begin{aligned} P[T \cup V \cup W \cup X] = {} & P[T] + P[V] + P[W] + P[X] \\ & - P[T \cap V] - P[T \cap W] - P[T \cap X] \\ & - P[V \cap W] - P[W \cap X] - P[V \cap X] \\ & + P[T \cap V \cap W] + P[T \cap V \cap X] \\ & + P[T \cap W \cap X] + P[V \cap W \cap X] \\ & - P[T \cap V \cap W \cap X] \end{aligned} \qquad 4.5.1$$

The terms for the expressions of equation 4.5.1 is given in equation 4.5.2.



*nodes 17, 18 are STPs; nodes 13-16 are intermediate SPs; all other nodes are SPs

Figure 4.5 : Hierarchical topology

$$P[T] = A_T$$
$$P[V] = A_V$$
$$P[W] = A_W$$
$$P[X] = A_X$$

$$\begin{aligned} P[T \cap V] = P[T]P[V/T] = {} & A_{N14} \cdot A_T A_{11L14} \cdot A_{11L17} A_{N16} \cdot A_T \\ & A_{17L16} \cdot A_{16L12} \end{aligned}$$
$$P[T \cap W] = P[T]P[W/T] = A_T A_{13L18} A_{N18} A_{18L15}$$
$$\begin{aligned} P[T \cap X] = P[T]P[X/T] = {} & A_T A_{16L12} A_{N14} A_{11L14} A_{N18} \cdot A_{16} \\ & A_{14L18} \cdot A_{18L16} \end{aligned}$$
$$\begin{aligned} P[V \cap W] = P[V]P[W/V] = {} & A_V A_{11L13} A_{N13} A_{13L18} A_{N18} A_{N15} \\ & A_{18L15} \cdot A_{15L12} \end{aligned}$$
$$P[V \cap X] = P[W]P[X/W] = A_V A_{14L18} A_{N18} A_{18L16}$$
$$\begin{aligned} P[W \cap X] = P[V]P[X/V] = {} & A_W A_{11L14} A_{N14} A_{14L18} A_{N16} \cdot \\ & A_{18L16} \cdot A_{16L12} \end{aligned}$$

$$\begin{aligned} P[T \cap V \cap W] = P[T]P[V/T]P[W/V,T] = {} & A_T A_{11L14} A_{N14} \cdot A_{14L17} \\ & A_{17L18} A_{N16} A_{16L12} A_{N18} \\ & A_{13L18} \cdot A_{18L15} \end{aligned}$$
$$\begin{aligned} P[T \cap V \cap X] = P[T]P[V/T]P[X/V,T] = {} & A_T A_{11L14} A_{N14} A_{14L17} \\ & A_{17L16} A_{N16} A_{16L12} A_{14L18} \\ & \cdot A_{18L16} A_{N18} \end{aligned}$$
$$\begin{aligned} P[T \cap W \cap X] = P[T]P[W/T]P[X/W,T] = {} & A_T A_{13L18} A_{N18} A_{18L15} \\ & A_{11L14} A_{N14} A_{14L18} \cdot A_{N16} \\ & A_{18L16} \cdot A_{16L12} \end{aligned}$$
$$\begin{aligned} P[V \cap W \cap X] = P[V]P[W/V]P[X/W,V] = {} & A_V A_{11L13} A_{N13} A_{13L18} \\ & A_{18L15} A_{N18} A_{N15} A_{15L12} \\ & A_{14L18} \cdot A_{18L16} \end{aligned}$$

$$\begin{aligned} P[T \cap V \cap W \cap X] = P[T]PV/T]P[W/V,T]P[X/W,V,T] = {} & \\ & A_T A_{11L14} A_{14L17} A_{N14} \\ & A_{17L18} A_{16L12} A_{N16} A_{13L18} \\ & A_{18L15} A_{N18} A_{14L18} A_{18L16} \\ & \cdot A_T A_{11L14} \cdot A_{11L17} \end{aligned} \qquad 4.5.2$$

The reliability for these routes can be given by equation 4.5.3 and can be obtained in the same way as the availability.

$$R(t) = R_T + R_V + R_W + R_X - R_{N14}R_{N16}R_T - R_{N18}R_T - R_{N14}R_{N18}R_{N16}R_T - R_{N13}R_{N16}R_{N15}R_V - R_{N18}R_V - R_W R_{N14}R_{N16} + R_T R_{N18}R_{N14}R_{N16} + R_T R_{N14}R_{N16}R_{N18} + R_V R_{N13}R_{N18}R_{N15} + R_T R_{N14}R_{N16}R_{N18} - R_T R_{N14}R_{N16}R_{N18} \quad 4.5.3$$

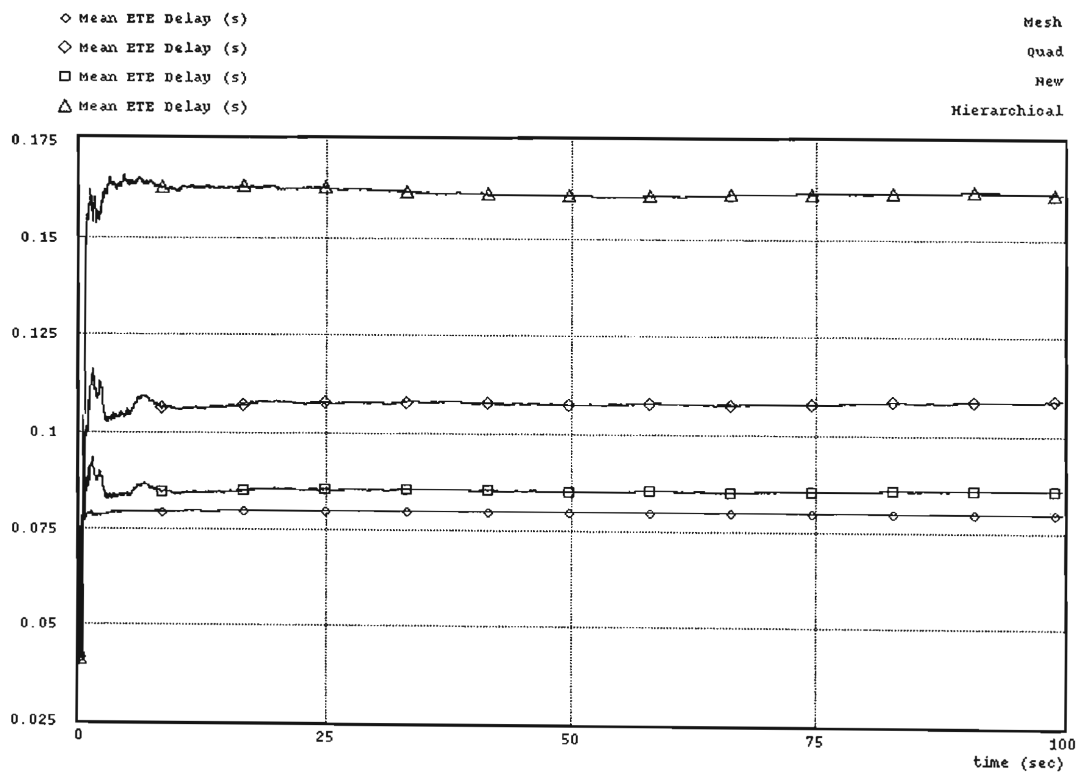
The availability and reliability can be calculated using Mathlab, where the MTTR and MTBF values were obtained from table 4.1. The following values were obtained for the hierarchical topology :-

- Availability - 525 542. 007 040 384 minutes/year
- Unavailability - 52. 992 959 616 065 24 minutes/year
- Reliability - 0. 992 432 616 143 1

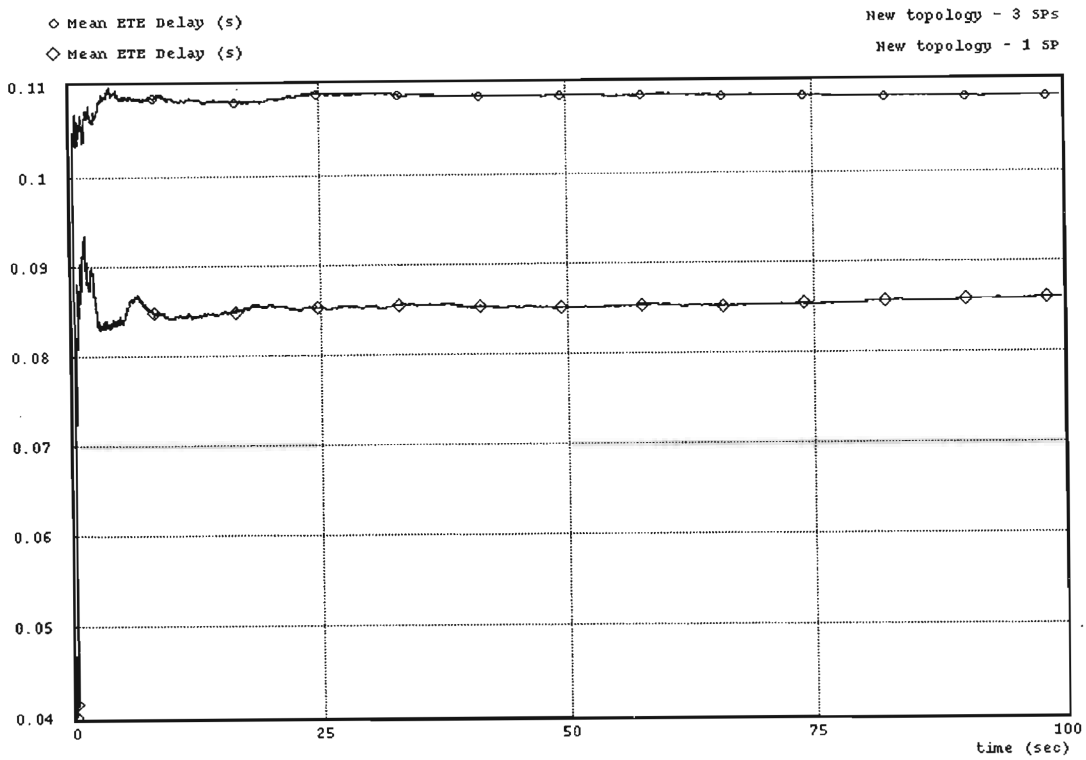
4.6 Signalling delays and the Kwa-Zulu Natal signalling network

Sections 4.1 - 4.5 looked at CCS7 topologies that have been used with success in other countries. These topologies were also analysed from an availability and reliability perspective that involved complex mathematics as the routes in a route set increased. A simpler technique requires the use of simulation software to show the delays existing in each of these networks. This technique is simple since the requirements of obtaining signalling delays have already been programmed in OPNET and can be applied to any of the networks discussed. The end to end delays of these networks appear in simulation 4.1, where the number of source SPs for the quad, hierarchical and new topologies is considered to be one. This simulation proves that the mesh and new structures have the best performance when considering end to end delays.

It should be noted that these delays corresponds to networks where the number of source SPs were taken to be one. For networks with more than one source SP, the delay can be expected to increase. As an example, consider the new topology of figure 4.4 where the number of source SPs were 3 (nodes 6, 12 and 13) and the number of destination SPs was 1 (node 7). If the end to end delay of this network is compared with a similar network consisting of a single origination/destination pair, the delay is larger for the network with more source SPs. This result can be seen in simulation 4.2. However, an increase in the number of SPs for all networks would still result in the mesh topology providing the least delays, followed by the new topology, etc.



Simulation 4.1 : Comparison of end to end delays.



Simulation 4.2 : Comparison of delays for different numbers of source SPs

Performance parameters such as availability, reliability and signalling delays can give a good indication of which networks to select for use in a particular signalling area. Another issue to be considered is the cost - the amount of capital required for replacing, modifying or upgrading the existing network structure. This requires the plans of the existing signalling network, so that suitable modifications can be made to transform the existing topology.

However, only the signalling point codes could be obtained for the Kwa-Zulu Natal area so that the entire signalling map would have to be constructed from these codes. The point codes obtained are as follows :

Durban New Germany NDNG:01,DPSU,NAT0,5-1-1-0

Durban New Germany NDNG:01,DPSU,NAT1,5-1-1-0

Durban Taj Mahal 1	NDTM:01,DPSU,NAT0,5-3-1-0
Durban Taj Mahal 1	NDTM:01,DPSU,NAT1,5-1-2-0
Durban Taj Mahal 2	NDTM:02,DPSU,NAT0,5-3-15-0
Amanzimtoti	NAMN:01,DSSU,NAT0,5-1-2-0
Chatsworth	NCHO:04,DSSU,NAT0,5-1-3-0
Durban Central 2	NDN+:03,DSSU,NAT0,5-1-4-0
Durban Central 1	NDN+:04,DSSU,NAT0,5-1-5-0
Durban North	NDNN:02,DSSU,NAT0,5-1-6-0
Greyville	NGVX:01,DSSU,NAT0,5-1-7-0
Isipingo	NIO+:04,DSSU,NAT0,5-1-8-0
Kloof	NKLO:02,DSSU,NAT0,5-1-9-0
Overport	NDVT:03,DSSU,NAT0,5-1-10-0
Phoenix 1	NPHX:01,DSSU,NAT0,5-1-11-0
Phoenix 2	NPHX:02,DSSU,NAT0,5-1-12-0
Pinetown	NPWN:02,DSSU,NAT0,5-1-13-0
Rossburgh	NRGH:03,DSSU,NAT0,5-1-14-0
Empangeni	NEPI:03,DSSU,NAT0,5-3-2-0
Ladysmith	NLYS:03,DSSU,NAT0,5-3-3-0
Pietermaritzburg	NPMB:06,DSSU,NAT0,5-3-4-0
Port Shepstone	NPS+:03,DSSU,NAT0,5-3-5-0
Newcastle	NNWC:03,DSSU,NAT0,5-3-6-0

If Newcastle is considered, the following hold true : NNWC = Kwa-Zulu Natal Newcastle; DSSU = digital secondary switching unit (in the case for New Germany and Taj Mahal, DPSU = digital primary switching unit); NAT0 = Kwa-Zulu Natal 0 (in the case for New Germany and Taj Mahal, NAT 1 = Kwa-Zulu Natal 1). The four digits X1-X2-X3-X4 at the end of the code represent the different NATx options in CCS7 signalling :

For NAT0 :

X1 = 5 stands for the province KZN

X2 = 1 stands for the Durban Metropolitan area (Overport, Durban North)

X2 = 3 stands for the "country" area (Ladysmith, Empangeni)

X3 = x stands for the different Exchanges (DSSUs)

X4 = 0 default value (not used)

For NAT1 :

X1 = 5 stands for the province KZN

X2 = 1 stands for the operator Telkom

X2 = 2 stands for the mobile operator MTN

X2 = 3 stands for the mobile operator Vodacom

X3 = x stands for the different Telkom exchanges or MSC (master station)

X4 = 0 default value (not used)

Using the above code, the following assumptions can be made :-

- The DPSUs can be related to transit exchanges (intermediate SPs) while the DSSUs can be related to local exchanges (source/destination SPs). It should be noted that it is a very rare occurrence to have subscribers working off DPSUs.

- For X2 = 3 (NAT0), this relates to country areas. However in Durban Taj Mahal, there are two DPSUs. The reason for Durban Taj Mahal having two DPSUs is as follows : In the past, due to the limited switching capabilities of the EWSD Siemens equipment, Telkom Kwa-Zulu Natal set up two co-located exchanges to handle the traffic volume. Although these two DPSUs have been replaced by one exchange, this one exchange is sometimes called by the older point codes.

- New Germany contains only one DPSU (NATO) for the entire Durban Area. All the DSSUs for X2=1 (NATO) are connected to the New Germany DPSU.
- Nat0 relates to all Telkom calls whereas Nat1 allows "inter-calls" between the different telecommunications operators such as VODACOM, MTN, etc.

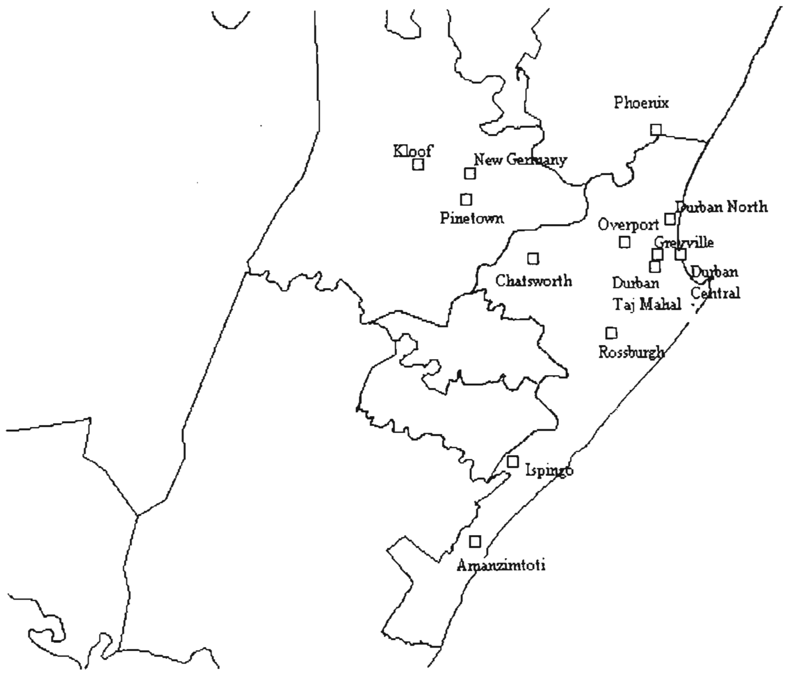
These assumptions listed out for these point codes would help in constructing the links between these SPs. The links are actually two trunks that carry signalling messages between the various SPs. In Kwa-Zulu Natal, there are no STPs. The next step was to obtain the coordinates for the location of these SPs. These coordinates could be accurately obtained from Computamaps in Cape Town and occur in the latitude - longitude format :

New Germany	30.872481	-	29.787614
Durban	31.030100	-	29.852300
Amanzimtoti	30.887000	-	30.046300
Greyville	31.010964	-	29.850823
Ispingo	30.906250	-	29.996290
Kloof	30.831665	-	29.789534
Overport	30.992807	-	29.831800
Phoenix	31.015100	-	29.704300
Pinetown	30.872000	-	29.807600
Rossburgh	30.980627	-	29.902842

Empangeni	28.556700	-	28.761300
Ladysmith	29.781800	-	28.556700
Pietermaritzburg	30.398800	-	29.620500
Port Shepstone	30.456800	-	30.742800
Newcastle	29.933700	-	27.744900

The next step was to plot these coordinates on a the map of Kwa-Zulu Natal. This was done on a student's geographical information system and is given on map 1 (entire Kwa-Zulu Natal) and map2 (zoom in of the Durban area).

Map 1 : Location of SPs in Kwa-Zulu Natal



Map 2 : Location of SPs in the Durban area

The SP codes and the locations give an impression of the structure of the existing signalling network. The codes give an overall picture of how the nodes are connected to one another. The expanded view of the signalling network can be seen in figure 4.6 (the scale of the province has been omitted). This network would have to be rearranged in order to distinguish if any characteristics are exhibited. The changed topology can be seen in figure 4.7 and shows the characteristics of the hierarchical topology. It can be clearly seen that this topology has no STPs and that the origination nodes, destination nodes and the intermediate SPs form the lowest level of the hierarchy.

For signalling between the different regions, the following route was chosen : X = Greyville – New Germany – Taj Mahal – Pietermaritzburg. The availability for this Kwa-Zulu Natal network's route set may be given as :

$$A_{SET} = A_X = A_G A_N A_T A_P \quad 4.6.1$$

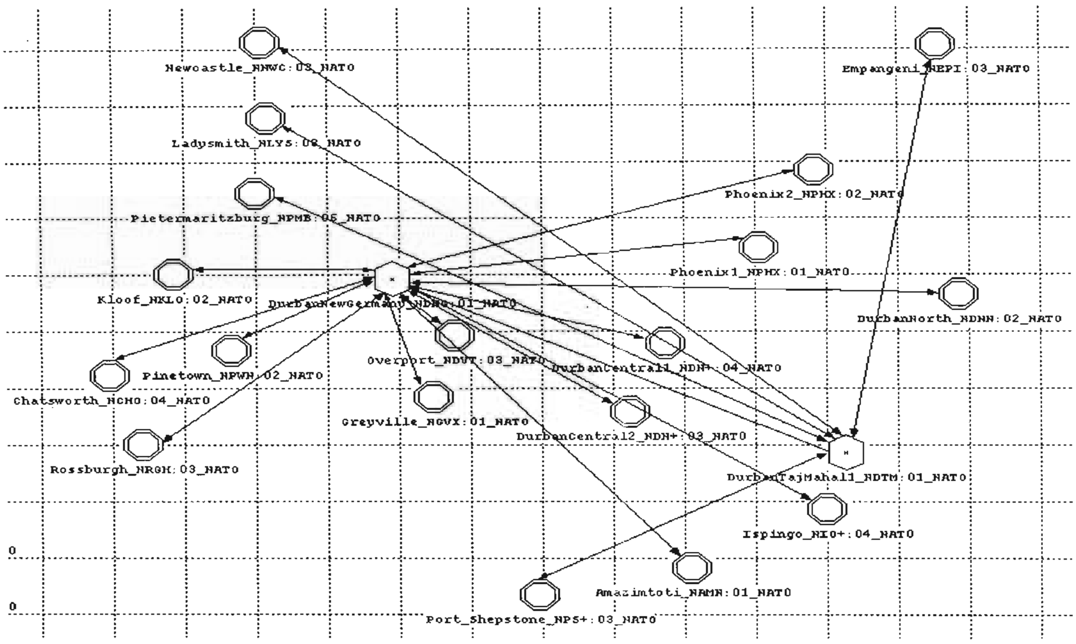


Figure 4.6 : Expanded view of Kwa-Zulu Natal's signalling network (out of scale)

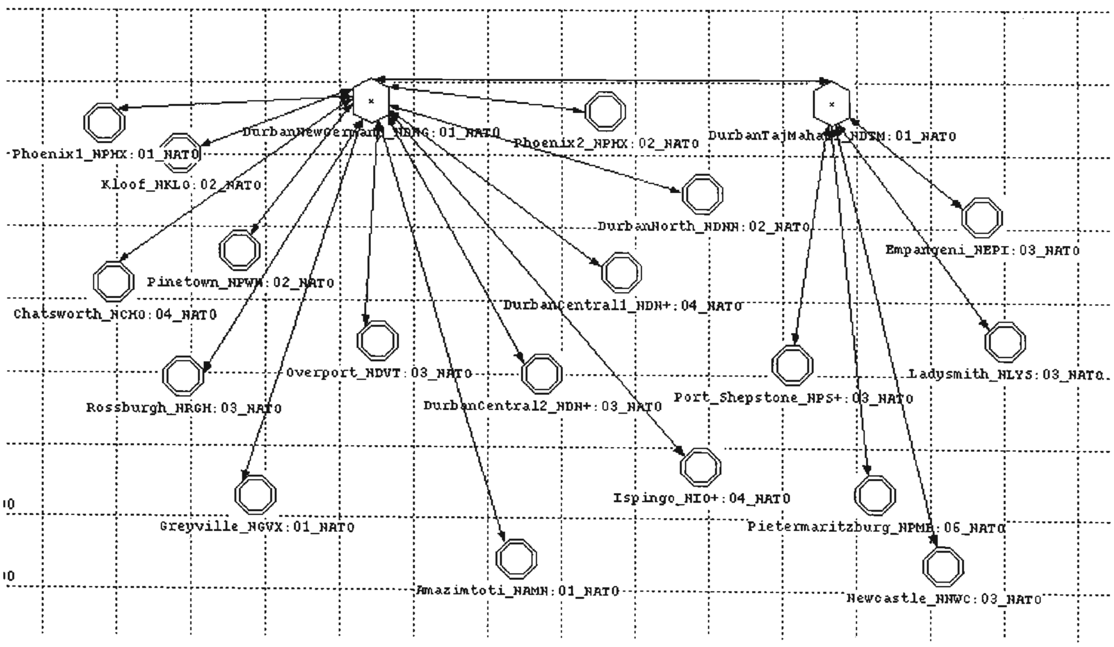


Figure 4.7 : Hierarchical view of Kwa-Zulu Natal's network

The reliability for this route set is given by :

$$R_{SET} = R_X = R_G R_N R_T R_P \qquad 4.6.2$$

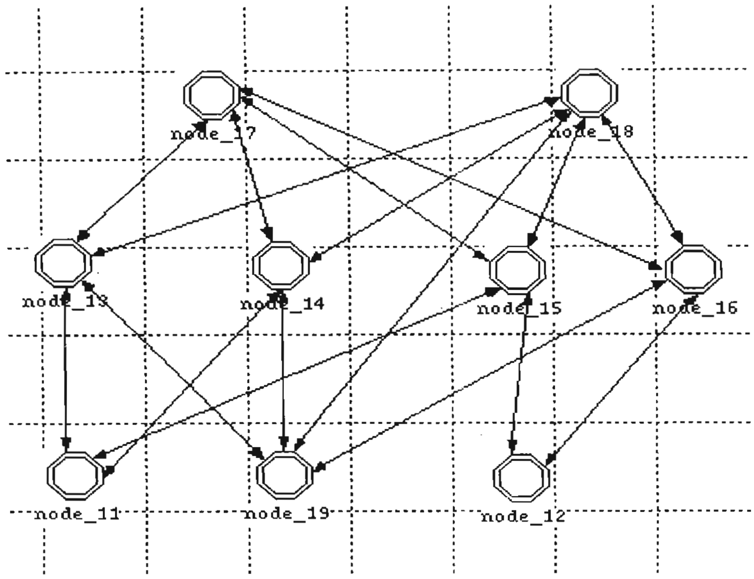
The availability, unavailability and reliability was calculated using Mathlab :-

- Availability = 525 588.000 102 739 4 minutes/year
- Unavailability = 11.999 897 260 568 10 minutes/year
- Reliability = 0.992 418 492 206 62

It should be noted that the Kwa-Zulu Natal signalling system has only one route in it's route set when considering signalling between any two network end points. This network does not exhibit characteristics of topological diversity in that redundancy of network elements are required i.e. network is not robust. From the results, the unavailability of any inter-regional signalling route set exceeds the ITU-T specification of 10 minutes/year. Further, the network does not meet the standard ITU-T requirement of a STP pair.

The use of a hierarchical topology for the existing Kwa-Zulu Natal's signalling network would result in lower costs and can be achieved efficiently. However, the ITU-T recommendations specify that the unavailability of a signalling route set should not exceed 10 minutes/year. The calculated unavailability for the hierarchical topology in section 4.5 is far greater than 10 minutes/year and therefore, this network cannot be used in its basic form.

An alternate form of this network can be specified where "short cut" links are used to minimise end to end delays. Figure 4.8 shows the basic hierarchical network with "short cut" links (these links occur between nodes 16 and 19 and nodes 18 and 19). Short cut links generally occur between nodes where the traffic loading is very high and connects SPs to intermediate SPs and STPs. In figure 4.8, nodes 11, 12 and 19 are SPs; nodes 13, 14, 15 and 16 are intermediate SPs and nodes 17 and 18 are STPs.



*nodes 17, 18 are STPs; nodes 13-16 are intermediate SPs; all other nodes are SPs

Figure 4.8 : Hierarchical topology with shortcut links

Using the network in figure 4.8, the following paths can be defined in table 4.6. These paths correspond to the four shortest routes between SPs 19 and 12.

Route Name	Path Followed (NX - node X; XLY - link connecting nodes X and Y; X,Y – integers)								
T	N19	19L14	N14	14L18	N18	18L15	N15	15L12	N12
V	N19	19L14	N14	14L18	N18	18L16	N16	16L12	N12
W	N19	19L18	N18	18L16	N16	16L12	N12		
X	N19	19L16	N16	16L12	N12				

Table 4.6 : Routes for hierarchical structure with short cut links

The availability for these routes can be described by equation 4.7.1 :

$$\begin{aligned} P[T \cup V \cup W \cup X] = & P[T] + P[V] + P[W] + P[X] \\ & - P[T \cap V] - P[T \cap W] - P[T \cap X] \\ & - P[V \cap W] - P[W \cap X] - P[V \cap X] \\ & + P[T \cap V \cap W] + P[T \cap V \cap X] \\ & + P[T \cap W \cap X] + P[V \cap W \cap X] \\ & - P[T \cap V \cap W \cap X] \end{aligned} \qquad 4.7.1$$

The terms for the expressions in equation 4.7.1 is defined in equations 4.7.2 :

$$\begin{aligned} P[T] &= A_T \\ P[V] &= A_V \\ P[W] &= A_W \\ P[X] &= A_X \end{aligned}$$

$$\begin{aligned} P[T \cap V] &= P[T]P[V/T] = A_{N16} \cdot A_T A_{16L12} \cdot A_{18L16} \\ P[T \cap W] &= P[T]P[W/T] = A_T A_{19L18} A_{N16} A_{18L16} A_{16L12} \\ P[T \cap X] &= P[T]P[X/T] = A_T A_{16L12} A_{N16} A_{19L16} \\ P[V \cap W] &= P[V]P[W/V] = A_V A_{19L18} \\ P[V \cap X] &= P[W]P[X/W] = A_V A_{19L16} \\ P[W \cap X] &= P[V]P[X/V] = A_W A_{19L16} \end{aligned}$$

$$\begin{aligned} P[T \cap V \cap W] &= P[T]P[V/T]P[W/V,T] = A_T A_{18L16} A_{N16} \cdot A_{16L12} A_{19L18} \\ P[T \cap V \cap X] &= P[T]P[V/T]P[X/V,T] = A_T A_{18L16} A_{N16} A_{16L12} \\ P[T \cap W \cap X] &= P[T]P[W/T]P[X/W,T] = A_T A_{19L18} A_{N16} A_{18L16} A_{19L16} A_{16L12} \\ P[V \cap W \cap X] &= P[V]P[W/V]P[X/W,V] = A_V A_{19L18} A_{19L16} \end{aligned}$$

$$\begin{aligned} P[T \cap V \cap W \cap X] &= P[T]PV/T]P[W/V,T]P[X/W,V,T] = \\ & A_T A_{18L16} A_{16L12} A_{N16} A_{19L18} A_{19L16} \end{aligned} \qquad 4.7.2$$

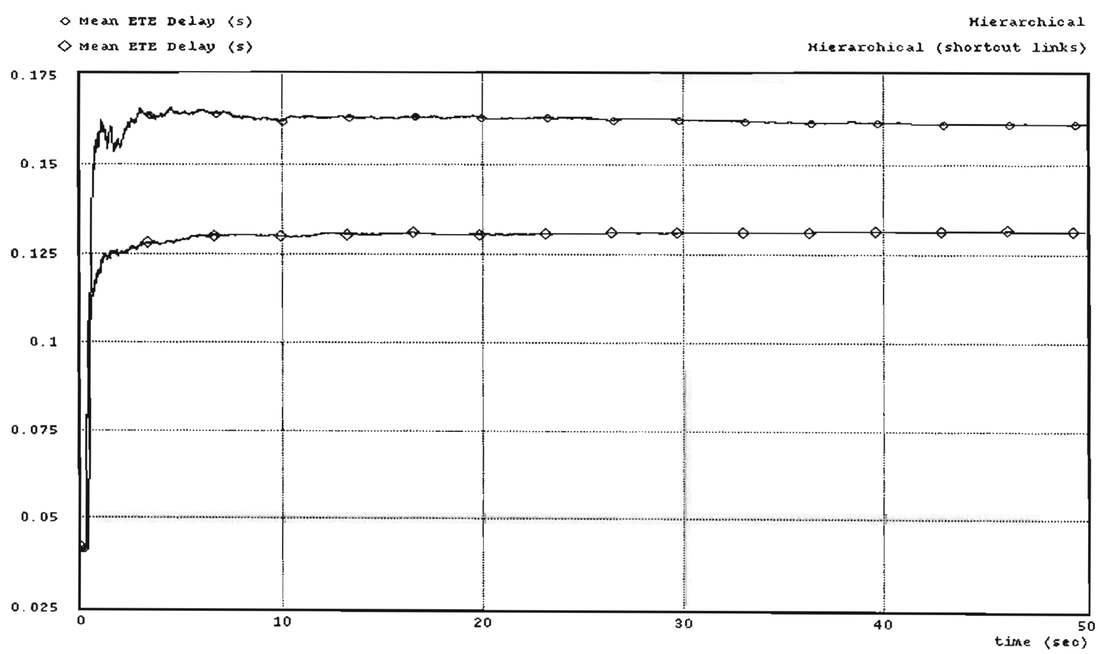
The reliability for the hierarchical structure with shortcut links is described by equation 4.7.3 and can be obtained in the same manner as the expression for the availability.

$$
\begin{aligned}
R(t) = {} & R_T + R_V + R_W + R_X - R_{N16}R_T - R_{N16}R_T - R_{N16}R_T \\
& - R_V - R_V - R_W + R_T R_{N16} \\
& + R_T R_{N16} + R_V + R_T R_{N16} \\
& - R_T R_{N16}
\end{aligned}
\qquad 4.7.3
$$

Using the values in table 1 and Mathlab to evaluate these equations, the following results were obtained :-

- Availability = 525 593. 996 158 910 3 minutes/year
- Unavailability = 6. 003 841 089 652 08 minutes/year
- Reliability = 0. 996 191 255 556 273

The unavailability for this route set is less than 10 minutes/year and meets the specifications of the ITU-T recommendations. Besides the availability and reliability being improved, the end-to-end delays for the hierarchical network with shortcut links are also less than the basic hierarchical structure (see simulation 4.3).



Simulation 4.3 : End to end delays for the different hierarchical topologies

The availability, unavailability, reliability and delays for the different topologies are shown in table 4.7. It can be seen that mesh topology gives the best results while the hierarchical structure can be improved with shortcut links.

	Availability (minutes/year)	Unavailability (minutes/year)	Reliability (1000 minutes)	End-to-end delays (s)
Mesh	525596.9999315081	3.00006849189982	0.99953216725586	0.08
Quad	525593.997566428	6.02433521863000	0.99619483465536	0.11
New	525594.0000171233	5.99998287676051	0.99620203381679	0.085
Hierarchical	525542.00704038400	52.99295961606524	0.99243261614310	0.165
Hierarchical (shortcut links)	525593.93996158910	6.00384108965208	0.996191255556273	0.13

Table 4.7 : Availability, reliability and delays for different topologies

The last factor to be considered would be the cost. As mentioned earlier, the hierarchical topology is best suited to the signalling system in Kwa-Zulu Natal as this network is basically hierarchical in nature. The mesh topology could also be applied to this region without affecting the existing signalling system, but only at a very high cost. If the 19 nodes in Kwa-Zulu Natal are meshed, the resulting network can be seen in figure 4.7. The result is a very impractical network where the cost would just be too high. Also, the implementation of such a network would take a lot of time and can only be achieved in the long term.

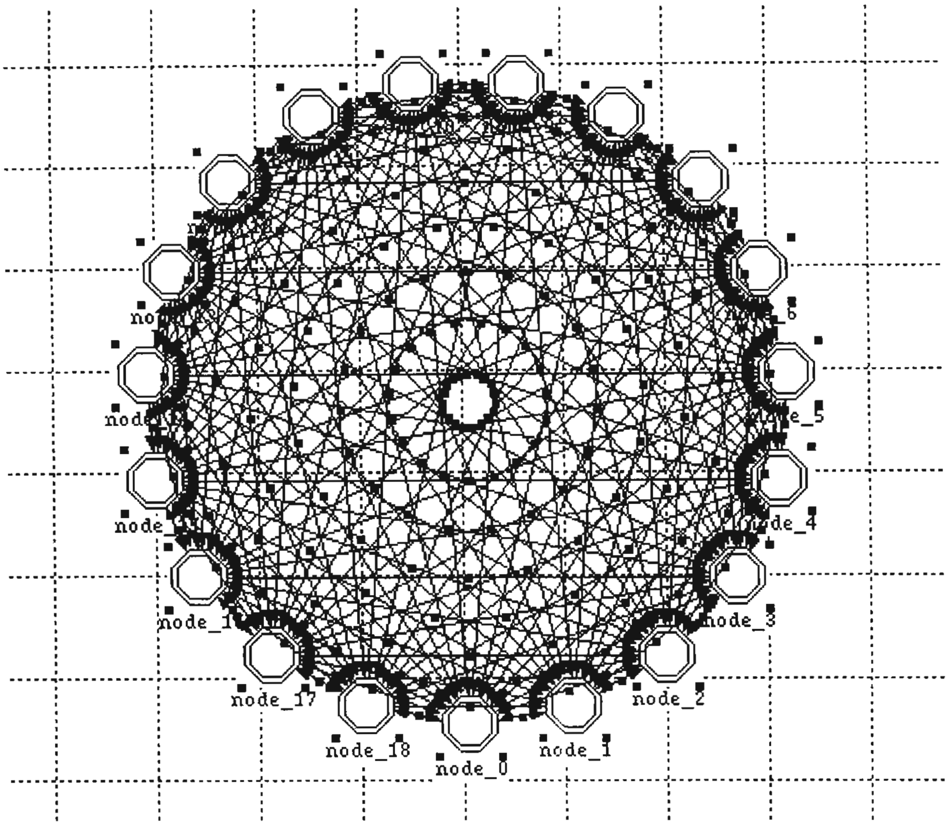


Figure 4.7 : Kwa-Zulu Natal's signalling system meshed

Therefore, the network in Kwa-Zulu Natal would have to be in a hierarchical form with shortcut links. The basic network can be see in figure 4.8. The network has the required redundancy in terms of SPs, intermediate SPs and links. Also, the required availability specifications have been met. The network only lacks STPs but this can be achieved at a national level.

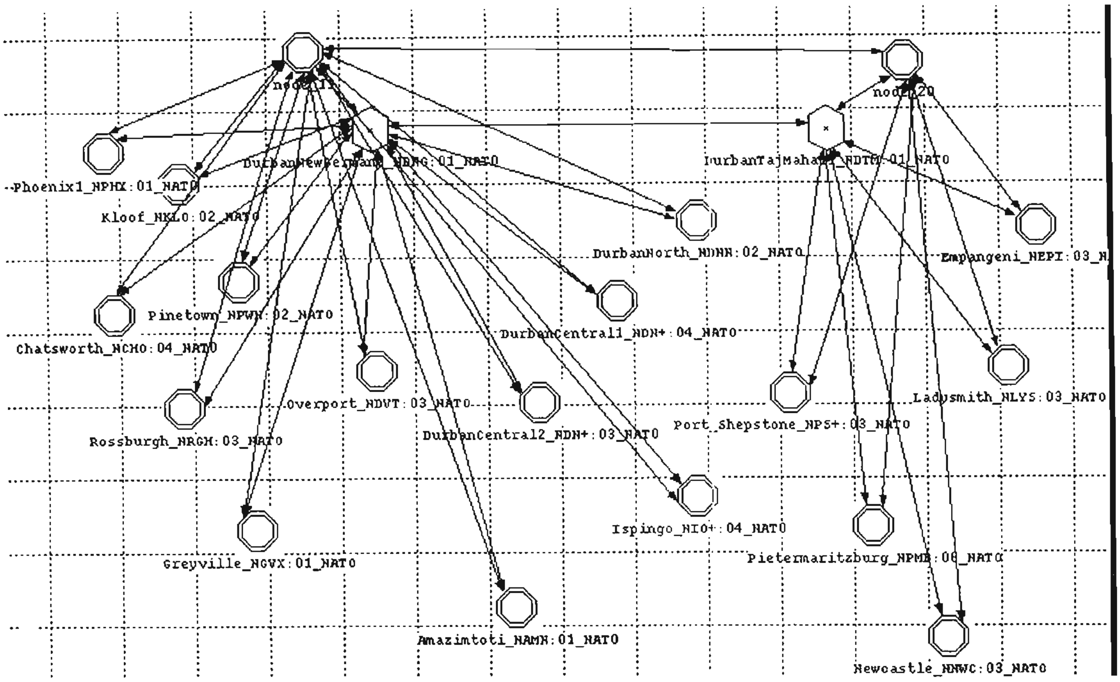


Figure 4.8 : Proposed Kwa-Zulu Natal's signalling network

The proposed network is identical to the hierarchical structure, except that there are no STPs on the highest level of the hierarchy. This top layer of the hierarchy usually represents the national level of signalling. Another issue would be the use of shortcut links. These can simply be added on and depends on the signalling traffic loads between origination and destination points.

5. Conclusion and Recommendations

This research focused on the restructuring of the Kwa-Zulu Natal signalling system. This required network selection by the comparison of the performance requirements of different topologies and the cost effective movement of the existing network to a target network in a practical and flexible way.

Performance parameters necessary in network selection are defined in chapter 2. These performance parameters can be classified as signalling delays and signalling network dependability. Signalling delays can be used to determine the end-to-end delays of different topologies while network dependability determines if the stringent availability, reliability and robustness criteria are met.

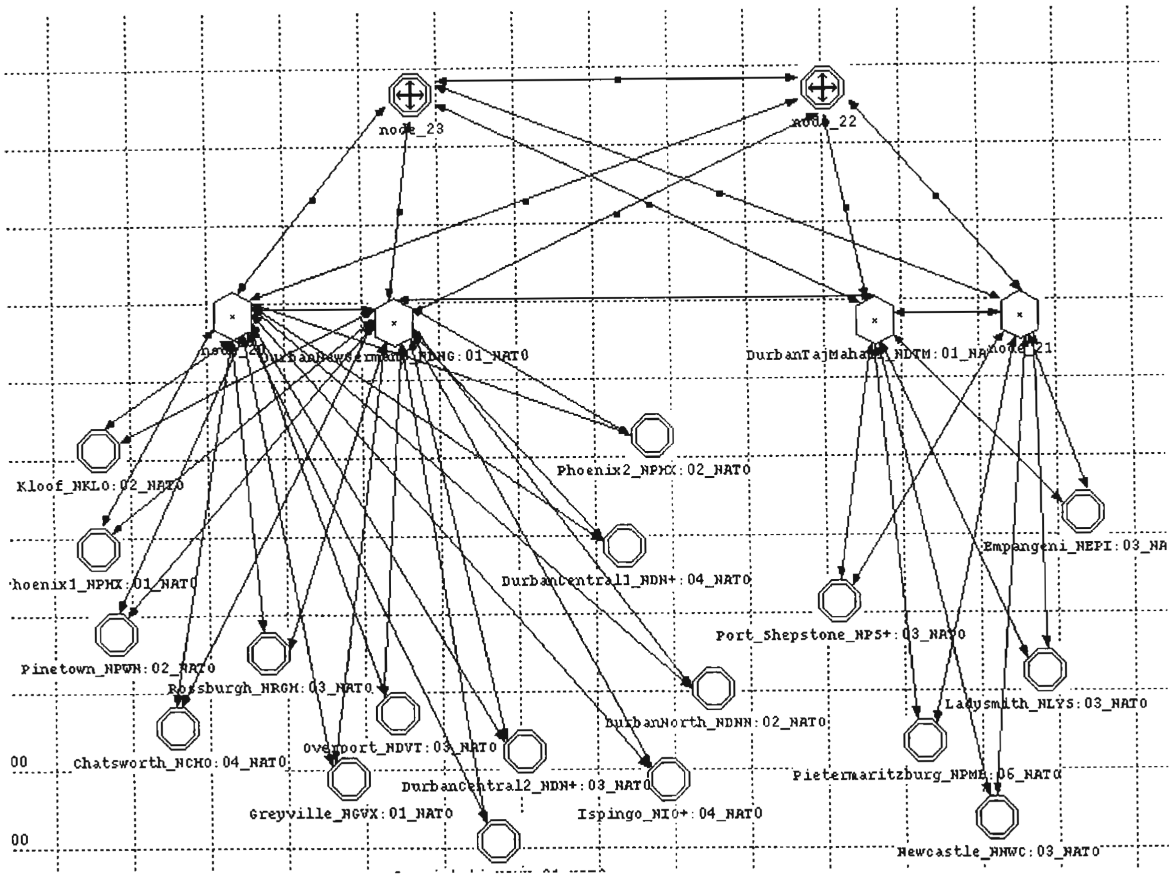
Chapter 3 described the model for the MTP and also showed the model implementation in OPNET. In this chapter, both the quad structure and a simple SP-SP relation were investigated to verify the operation of CCS7 within OPNET. The software package was used in chapter 4 to simulate the end to end delays of the signalling system's topologies.

The quality of service parameters and the simulations from OPNET were used to select the CCS7 network with the best performance characteristics. It was found that the mesh structure had the best analytical and simulated results but was not practical for large numbers of signalling points.

The hierarchical structure was the most cost-effective but this topology did not meet with the availability requirements. Shortcut links were introduced to improve the availability between routes and this seemed a viable solution since the existing Kwa-Zulu Natal signalling system was a hierarchical topology, and duplicated elements as well as excess links could just be added on without affecting operation of the existing the network.

The hierarchical topology meets the stringent standards set by the ITU-T in that there is sufficient redundancy to support the network in the event of a route failure. In addition, this network prevents cyclic routing, as in the case of the quad structure, and can be applied to the existing Kwa-Zulu Natal signalling system without changing the existing network's skeleton. Due to the advent of new services and simultaneous increase in traffic, shortcut links can always be added to this hierarchical structure to support new enlarged loads.

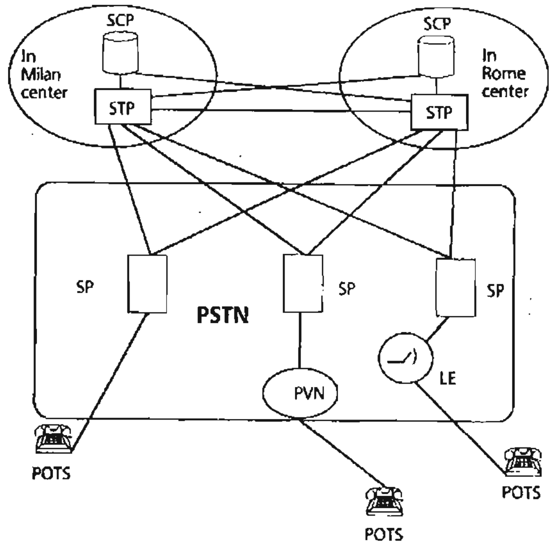
Another issue to consider would be the presence of duplex STPs within this signalling system. As an example, consider the signalling network being deployed in North America and Telstra's CCS7 topology (refer to sections 4.3 and 4.4). Both these networks have four STPs that are mated on a pairwise basis. Figure 5.1 shows the Kwa-Zulu Natal's signalling system with SPs, intermediate SPs and duplex STPs that form the highest level of the hierarchy.



* nodes 22, 23 are STPs

Figure 5.1 : Hierarchical signalling network with STPs

The highest level of the hierarchy corresponds to STPs at a national level. The STPs at a national level connect to the signalling end points or the intermediate signalling points of all provincial signalling networks. This is typical of topologies being currently employed in Europe, where mated STPs form the highest level of signalling networks. In figure 5.2, the telephone service of Telecom Italia's CCS7 network can connect either to a intermediate SP or a signalling end point. The network has two national STPs at the Rome and Milan centers that connect the intermediate SPs and the service control point (SCP), [38].



- POTS - plain old telephone service; PVN - private virtual network; PSTN - public switching telephone network; SCP - service control point; LE - local exchange.

Figure 5.2 : Telecom Italia's CCS7 network

In conclusion, the existing Kwa-Zulu Natal signalling network should be hierarchical in nature with shortcut links for selected routes. There should be sufficient redundancy in signalling elements so that in the event of route failure, an alternate route could handle the failed route's traffic. In this context, a route should be dimensioned to handle twice its load. Finally, it is recommended that the network architecture resemble those being currently employed in Europe where STPs form the highest level of the hierarchical structure and have national functionality.

6. REFERENCES

[1] S. Steven : 'Performance Evaluation of Signalling System No. 7'. University of Stellenbosch, August 1994, pp. 1 - 11.

[2] J. Sole Gil : 'Modelling the Common Channel Signalling Network for Planning the Structure of a Target Network'. *ITC 12th International Teletraffic Congress*, Torino, June 1988, pp. 2.4A.2.1 – 2.4A.2.8.

[3] 'Guidelines for Implementing a Signalling System No. 7 Network'. *CCITT Handbook*, Geneva, 1991, pp. 1 - 36.

[4] 'Message Transfer Part, Recommendation Q706'. *CCITT Blue Book*, vol VI, Fascicle VI.7, 1988, pp. 1 - 36.

[5] R. A. Skoog : 'Engineering Common Channel Signalling Networks for ISDN'. *ITC 12th International Teletraffic Congress*, Torino, 1988, pp. 2.4A.1.1 – 2.4A.1.7.

[6] R. A. Skoog : 'Study of Clustered Arrivals and Processes'. *Teletraffic and Datatraffic in a Period of Change*, 1991, pp. 61 -66.

[7] L. Rheeder : 'A Comparison between the number of Message Signal Units Sent and the Number of Voice Calls Carried in Telkom SA's Network'. *Regional International Teletraffic Conference*, South Africa, 1995, pp. A44 - 53.

[8] V. V. Karmarkar : 'Assuring SS7 Dependability : A Robustness Characterization of Signaling Network Elements'. *IEEE Journal on Selected Areas in Communications*, vol 12, no. 3, April 1994, pp. 475 - 489.

[9] N. Marlow and G. Brush : 'Assuring the Dependability of Telecommunications Networks and Services'. *IEEE Network Magazine*, January 1990, pp. 29 - 33.

[10] G. L. Steyn, H. E. Hanrahan : 'Signalling System No. 7 Availibility, Reliability and Dependability : A Simplified Approach'. *Regional International Teletraffic Conference*, South Africa, 1995, pp. 69 - 81.

[11] S. M. Campanholi , L. F. Cetolo, E. L. Ursini, A. V. Vinhas, J. M. Souza : 'Quality Measures of the CCITT Signalling System No. 7 - A Case Study'. Telebras, Research and Development Centre, pp. 111 - 119.

[12] A. C. Steenhuisen, R. Adams : 'Message Routing in the C7 Signalling Network'. *Trends in Telecommunications*, vol 9, no. 2, pp. 18 -24.

[13] M. M. Mostrel : 'Issues on the Design of Survivable Common Channel Signalling Networks'. *IEEE Journal on Selected Areas in Communications*, vol 12, no. 3, April, 1994, pp. 526 - 532.

[14] 'Hypothetical Signalling Reference Connection, Recommendation Q709'. *CCITT Blue Book*, vol VI, fascicle VI.7, 1988, pp. 1 - 12.

[15] 'Signalling Network Structure, Recommendation Q705'. *CCITT Blue book*, vol VI, fascicle VI.7, 1988, pp. 1 - 22.

[16] R. L. Freeman : 'Reference Manual for Telecommunications Engineering'. 2nd Edition, John Wiley and Sons, 1994, pp. 138 - 158.

[17] 'Signalling Network Functions and Messages, Recommendations Q704'. *CCITT Blue Book*, vol VI, fascicle VI.7, 1988, pp. 1 - 24.

[18] D. R. Manfield, G. Millsteed, M. Zukerman : 'Congestion Controls in SS7 Networks'. *IEEE Communications Magazine*, June 1993, pp. 50 - 57.

[19] 'ABC of the Telephone', *Traffic series,* pp. 1 - 25.

[20] O. Espvik : 'Teletraffic'. *Telektronikk.* March 1995, pp. 3 - 23.

[21] Warren G : 'Teletraffic Engineering 1996 - A Queueing Theory Primer'. University of Pretoria. 1996, pp. 21 - 27.

[22] J. van Duuren, P. Kastelein, F. C. Schoute : 'Telecommunication Services and Networks'. Addisdon-Wesley, 1993, pp. 107 - 119.

[23] L. Kleinrock : 'Queueing Systems - Volume 1 : Theory'. John Wiley and Sons, 1975, pp. 3 - 226.

[24] G. Williams : 'Modelling and Performance Evaluation of a Multi-Layered Signalling Network Based on the CCITT No.7 Specification'. *ITC 12th International Teletraffic Congress*, Torino, 1988, pp. 2.4A.3.1 – 2.4A.3.11.

[25] I. Chin, H. Cheong : 'A Simulation of the Level 2 of the CCITT CCS7'. *A.T.R.*, Vol. 18, No. 2, 1984, pp. 19 - 27.

[26] 'Functional Description of the Message Transfer Part (MTP) of Signalling System No. 7 Q701'. *CCITT Blue Book*,vol VI, fascicle VI.7, 1988, pp. 1 - 20.

[27] Bijan Jabbari : 'Routing and Congestion Control in Common Channel Signalling System No. 7'. *Proceedings of IEEE,* Vol. 80, No. 4, April 1994, pp. 607 – 617.

[28] T. Russel : 'Signaling System #7'. McGraw-Hill, Inc., pp. 61 – 115.

[29] 'Open Service Node : A Service Control System for Intelligent Networks'. White Paper, Intellitel Communications Ltd, pp. 11 - 15.

[30] J. U You, M. Y. Chung, T. S. Kim, D. K. Sung, B. D. Choi, K. S. Song : 'Development of a SS7 Simulator and Its Applications'. *7th International Network Planning Symposium*, Sydney, Austrailia 1996, pp. 95 - 100.

[31] M. Bafutto, P. J. Kuhn, G. Williams : 'Capacity and Performance Analysis of Signalling Networks in Multivendor Enviroments'. *IEEE journal on selected areas in communications*, Vol. 12, No. 3, 1994, pp. 490 - 500.

[32] D. R. Manfield, G. K. Millsteed, M. Zukerman : 'Performance Analysis of SS7 Congestion Controls Under Sustained Overload'. *IEEE journal on selected areas in communications*, Vol. 12, No. 3, 1994, pp. 405 - 414.

[33] 'OPNET Modelling Manual'. Release 3.0, Mil 3, Inc, Vol 2.1 : pp. Pst 61 – 115, Bpt 3 – 46; Vol 2.1 : Ntdef 37 – 48, Comec 38 – 42; Vol 6 : Id 3 – 13, Ima 3 – 19, Intrpt 3 - 48, Pk 3 –78; Vol 7 : Rte 3 –108, Sim 3 –12, Stat 3 –12, Strm 3 –16, Td 3 – 16, Topo 3 –20, Const 3 -19.

[34] 'Guidelines for SS No. 7 Network Design and Dimensioning'. Chapter 3, *CCITT recommendations,* pp 17 - 30.

[35] A. R. Modarressi, R. A. Skoog : 'An Overview of Signalling System No. 7'.Invited paper, *IEEE,* pp. 590 - 604.

[36] L. Krauss, G. Rufa : 'On the Design of a Hierarchical SS7 Network : A Graph Theoretical Approach'. *IEEE Journal on Selected Areas in Communications*, Vol. 12, No. 3, April 1994, pp 468 - 474.

[37] T. Hosaka, S. Kano, K. Kitami : 'New Common Channel Signaling System : An Overview'. *Review of the Electrical Communication Laboratories*, Vol. 28, Numbers 1-2, Jan-Feb 1980, pp. 50 - 61.

[38] R. S. Ramlakan, S. H. Mneney : 'Study of Performance Parameters Associated with Common Channel Signalling System No. 7'. *Teletraffic Systems Engineering Seminar 1996*, South Africa, University of Natal, 2 – 4 September 1996, pp 270 – 276.

[39] R. S. Ramlakan, S. H. Mneney : 'A Study of the Transmission Buffer within Common Channel Signalling System No. 7'. *Teletraffic – ITC Mini Seminar on Multimedia Networks,* South Africa, University of Rhodes, 1997, pp. 295 - 303.

Appendix

Appendix A

OPNET

This section describes the processes used in OPNET. Appendix A1 shows the OPNET protocols used for the pipeline stage. Appendix A2 shows the process model and the required software that modifies the existing error correction method to suit that of CCS7. Appendix A3 shows the process model for the static distributed routing algorithm. In this case, only the modifications are shown and the existing software can be obtained from OPNET. The generation of the address distribution for packets as well as the address allocation can be found in A4 while the sink process occurs in A5. It should be noted that the end to end delay statistics as well as the packet count can be found in A5.

Appendix A1 : MTPL1 and MTPL2 pipeline stage models

OPNET supports the modelling of point - to point links between communication nodes with procedures that compose a four stage pipeline which implements the transfer of data when transmissions begin. These procedures are shown in table A1.

"txdel" represents the transmission delay stage and is represented by the "dpt_txdel" (duplex point - to - point transmission delay) model. The result of this stage is a single value that represents the transmission delay for the packet. This stage is based on the bit rate of links and the lengths of packets.

“propdel” represents the propagation delay stage and is represented by the dpt_propdel (duplex point - to - point propagation delay) model. This stage represents the duration separating the time from when the first bit is transmitted till this bit is received at a destination node.

“closure” represents the closure stage and is represented by the “dpt_closure” (duplex point - to - point closure) model. This stage simply determines if there is connectivity between a receiver and transmitter for packets.

“coll” represents the collison model and is represented by the “dpt_coll” (duplex point - to - point collison) model. This stage determines if there is a collison or not - the presence of several packets simultaneously at a receiver.

“error” represents the error allocation stage and is represented by the “dpt_error” (duplex point - to - point error allocation) model. This stage allocates errors for each transmitted packet.

“ecc” represents the error correction stage and is represented by the “dpt_ecc” (duplex point - to - point error correction) model. The result of this stage is to determine the acceptability of this packet at the receiver.

Attribute	Model name
txdel model	Dpt_txdel
closure model	Dpt_closure
propdel model	dpt_propdel
coll model	Dpt_coll
error model	Dpt_error
ecc model	Dpt_ecc

Table A1 : Models used for different stages in pipeline

Appendix A2 : MTPL2 modified error correction process model

The following diagram (figure A2) demonstrates error correction process. The "init" state represents the initialisation state where variables and statistics are initialised at the beginning of a simulation. During the simulation, the process rests at the "idle" state. On a stream interrupt (a packet being received), the process moves to the "rcv" state where the error correction is performed. The process then moves to the "idle" state and the process repeats itself.

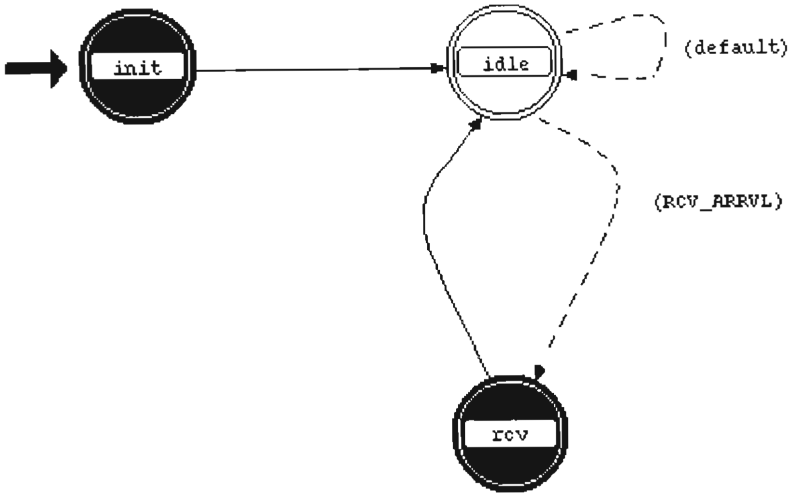


Figure A2 : Error correction process

Appendix A2.1 : Header block for error correction process

```
/* packet stream definitions */

# define RCV_IN_STRM 0

# define XMT_OUT_STRM 0

/* transition macros */

#define SRC_ARRVL ( op_intrpt_type () == OPC_INTRPT_STRM && \
                    op_intrpt_strm () == SRC_IN_STRM )

#define RCV_ARRVL ( op_intrpt_type () == OPC_INTRPT_STRM && \
                    op_intrpt_strm () == RCV_IN_STRM )
```

Appendix A2.2 : Temporary variables for error correction

```
Packet *pkptr;

double ete_delay;

int n, q, num_errors;
```

Appendix A2.3 : Init software for error correction process

```
pk_count = 0;

n = 0;

q = 0;

num_errors = 0;

pk_cnt_stathandle = op_stat_reg ("packet count", OPC_STAT_INDEX_NONE,

OPC_STAT_LOCAL);
```

Appendix A2.4 : Rcv software for error correction process

```
pkptr = op_pk_get (RCV_IN_STRM);

num_errors=op_td_get_int(pkptr, OPC_TDA_PT_NUM_ERRORS);

n = n + 1;


{if (num_errors > 0)

{num_errors = 0;
```

```
q = q + 1;
op_pk_send_delayed (pkptr, XMT_OUT_STRM, 0.120);
{n = 0;
q = 0;}}
{if (n > 0)
{pk_count = pk_count + 1;
op_stat_write (pk_cnt_stathandle, pk_count);
op_pk_send_delayed (pkptr, XMT_OUT_STRM, 0.040);}}
```

Appendix A3 : MTPL3 shortest path routing process

Figure A3 shows the OPNET defined routing process diagram. In the "wait" state, the process examines an incoming packet and moves to that state depending on the packet type.

"proc_arp" reads an incoming address resolution packet. The address resolution packet defines a direct route from a sending node to a receiving node. If this route is shorter than any existing route in the node's routing table, this route is added to the routing table. A route advertisement packet is then transmitted to nodes in the network defining this new short route.

"proc_ad" reads the incoming route advertisement packet. If this advertised route is better that any other route in the routing table or if this route does not exist, then it is added to the routing table.

“proc_data” reads the final destination of the packet and compares it with the node destination. If these destinations match, the packet is decapsulated and sent to the UPs.

“encap” encapsulates the packet from the UP in a data packet and sets the packets destination address. The process then enters the “proc_data” state.

The failure and recover modifications as well the detination node modification can be found in the header block and the “init state”.

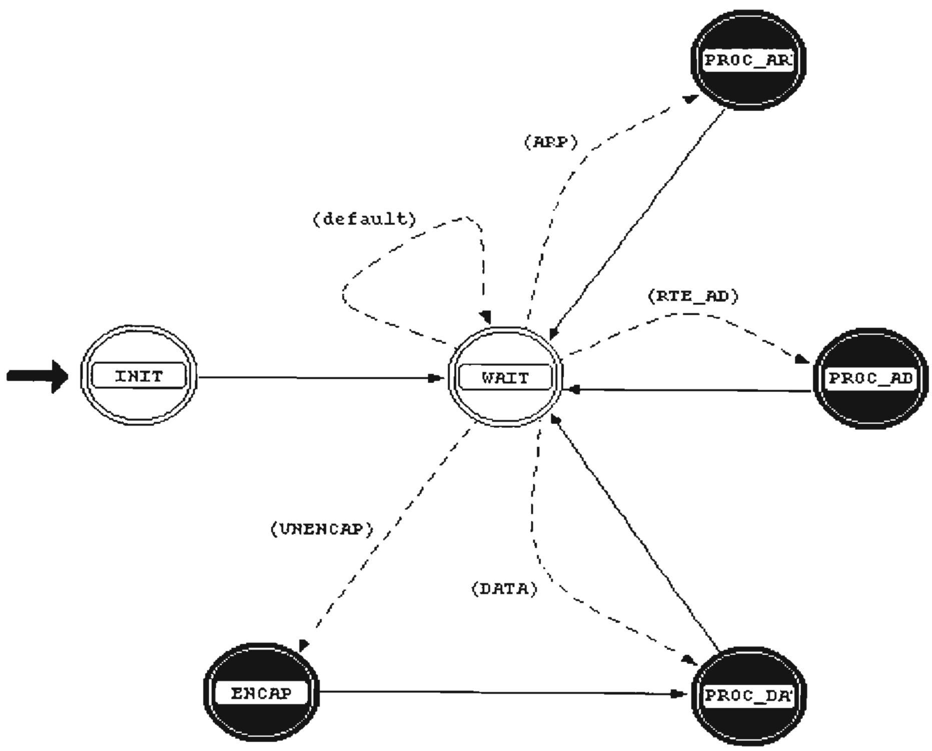


Figure A3 : Process for routing protocol

Appendix A3.1 : Header block for routing process

/*Other expressions are defined in OPNET and have been omitted */

```
void fail_recov_proc ();
```

Appendix A3.2 : Function block for routing process

```
/*Other functions and procedures are defined in OPNET and have been omitted */
void
fail_recov_proc (state_ptr, code)
char* state_ptr;
int code;
{Objid nd_objid; Objid node_id; Objid object_id;
 int object_type;
Topology*      rte_topo_ptr;
FIN (fail_recov_proc (state_ptr, code));
nd_objid = op_topo_parent (op_intrpt_source ());
object_id = op_intrpt_source();
object_type = op_id_to_type (nd_objid);
if (code == 1)
{ {op_ima_obj_attr_set(nd_objid,"condition",OPC_BOOLINT_ENABLED);  }
1,op_id_from_name(op_topo_parent (op_id_self ()), OPC_OBJTYPE_PROC, "sdr")); */
}
if (code == 0)
{{ printf("t");
op_ima_obj_attr_set(nd_objid,"condition",OPC_BOOLINT_DISABLED);
```

```
printf("t");}}
FOUT }
```

Appendix A3.3 : Temporary variables for routing process

```
/*Other temporary variables are defined in OPNET and have been omitted */
Objid                   object_id; nd_objid; module_id;
Topology*               rte_topo_ptr;
int FAIL_CODE=0, RECOV_CODE=1;
double fail_time, recover_time;
```

Appendix A3.5 : Init software for routing process

```
/* Register statistics to obtain handles to them. */
ads_rcvd_stathndl  = op_stat_reg ("Route Advertisements Received",
OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
ads_sent_stathndl  = op_stat_reg ("Route Advertisements Sent",
OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
cost_avg_stathndl  = op_stat_reg ("Cost", OPC_STAT_INDEX_NONE,
OPC_STAT_LOCAL);
num_routes_stathndl = op_stat_reg ("Number of Routes", OPC_STAT_INDEX_NONE,
OPC_STAT_LOCAL);
/* Set up statistics and debugging information. */
```

```
sdr_trace_active = op_prg_odb_ltrace_active ("sdr");
op_stat_write (ads_rcvd_stathndl, num_ads_rcvd = 0);
op_stat_write (ads_sent_stathndl, num_ads_sent = 0);
op_stat_write (cost_avg_stathndl, rte_cost_total = 0.0);
op_stat_write (num_routes_stathndl, 0);


/* Read the local node address. */
loc_mod_objid = op_id_self ();
loc_node_objid = op25_topo_parent (loc_mod_objid);
/* if (op_ima_obj_attr_get (loc_mod_objid, SDRC_ATTR_ADDR, &loc_node_addr) ==
OPC_COMPCODE_FAILURE) */

/* sdr_error ("Unable to get address from attribute.");*/
/* op_ima_obj_attr_get (loc_mod_objid, SDRC_ATTR_ADDR, &loc_node_addr); */
/* Assign the source node address. */
/* if (loc_node_addr != SDRC_AUTO_ASSIGNMENT) */
/*      sdr_error ("Invalid source node address."); */
/* else */
/* loc_node_addr = source_addr; */

op_ima_obj_attr_get(loc_mod_objid, "Source Address",&loc_node_addr);
```

```
/* loc_node_addr = source_addr; */
/* Increment the global address assignment counter. */
/* source_addr++;*/

if (sdr_trace_active)
{sprintf (msg_str, "Initializing node (%d):", loc_node_addr);
op_prg_odb_print_major (msg_str, OPC_NIL);}
/* Set up routing and interface tables. */
route_table = op_prg_list_create ();
interface_table = op_prg_list_create ();
if (route_table == OPC_NIL || interface_table == OPC_NIL)
        sdr_error ("Unable to allocate route or interface table.");
        num_routes = 0;
        num_interfaces = 0;
num_out_assoc = op_topo_assoc_count (loc_mod_objid, OPC_TOPO_ASSOC_OUT,
OPC_OBJTYPE_STRM);
for (i = 0; i < num_out_assoc; i++)
{strm_objid = op_topo_assoc (loc_mod_objid, OPC_TOPO_ASSOC_OUT,
OPC_OBJTYPE_STRM, i);
if (strm_objid == OPC_OBJID_INVALID ||
op_ima_obj_attr_get (strm_objid, "src stream", &strm) ==
OPC_COMPCODE_FAILURE)
{sdr_error ("Unable to get source stream index of attached output stream.");}
```

```
if (strm != 0)
{loc_mac_objid = op_topo_assoc (strm_objid, OPC_TOPO_ASSOC_OUT,
OPC_OBJMTYPE_MODULE, 0);
if (loc_mac_objid == OPC_OBJID_INVALID)
sdr_error ("Unable to get object ID of attached MAC object.");
if (op_id_to_type (loc_mac_objid) == OPC_OBJTYPE_PT_TX)
{/* This output leads to a point-to-point transmitter. */
/* Determine the cost of the associated link, if any.  */
if (op_topo_assoc_count (loc_mac_objid, OPC_TOPO_ASSOC_OUT,
OPC_OBJMTYPE_LINK) > 0)
{/* Get the object ID of the attached link. */
link_objid = op_topo_assoc (loc_mac_objid, OPC_TOPO_ASSOC_OUT,
OPC_OBJMTYPE_LINK, 0);
if (link_objid == OPC_OBJID_INVALID)
sdr_error ("Unable to get object ID of point-to-point link through connected
transmitter.");
/* Create and initialize a new interface record and add it to the table. */
intf_ptr = (SdrT_Intf *) op_prg_mem_alloc (sizeof (
{SdrT_Intf));
if (intf_ptr == OPC_NIL)
sdr_error ("Unable to allocate interface record.");
intf_ptr->strm = strm;
intf_ptr->type = SDRC_INTF_TYPE_POINT_TO_POINT;
```

```
op_prg_list_insert (interface_table, intf_ptr, OPC_LISTPOS_TAIL);
num_interfaces++;
/* Get the link cost. */
if (op_ima_obj_attr_get (link_objid, "cost", &intf_ptr->cost) ==
OPC_COMPCODE_FAILURE)
sdr_error ("Unable to get point-to-point link cost from attribute.");
if (sdr_trace_active)
{sprintf (msg_str, "Output (%d): point-to-point link, cost (%g)", i,intf_ptr->cost);
op_prg_odb_print_minor (msg_str, OPC_NIL);}
/* Send an Address Resolution Packet.  Give a dummy value for the interface address. */
pk_ptr = op_pk_create_fmt ("sdr_arp");
if (pk_ptr == OPC_NIL ||
op25_pk_nfd_set (pk_ptr, "node_addr", loc_node_addr) ==
OPC_COMPCODE_FAILURE ||
op25_pk_nfd_set (pk_ptr, "intf_addr", 0) == OPC_COMPCODE_FAILURE)
{sdr_error ("Unable to create or initialize address resolution packet.");}
op_pk_send (pk_ptr, i);}
else if (sdr_trace_active)
{sprintf (msg_str, "Output (%d): unconnected point-to-point transmitter", i);
op_prg_odb_print_minor (msg_str, OPC_NIL);}}
else
{/* All link types besides point-to-point require some sort of medium access control. */
```

```
if (op_ima_obj_attr_get (loc_mac_objid, "process model", mac_model) ==
OPC_COMPCODE_FAILURE)
sdr_error ("Unable to get MAC process model name from attribute.");
if (!strcmp (mac_model, SDRC_INTF_ETHERNET_ATTR_MODEL))
{strm_objid = op_topo_assoc (loc_mac_objid, OPC_TOPO_ASSOC_OUT,
OPC_OBJTYPE_STRM, 0);
if (strm_objid == OPC_OBJID_INVALID)
loc_tx_objid = op_topo_assoc (strm_objid, OPC_TOPO_ASSOC_OUT,
OPC_OBJMTYPE_MODULE, 0);
if (loc_tx_objid == OPC_OBJID_INVALID)
if (op_topo_assoc_count (loc_tx_objid, OPC_TOPO_ASSOC_OUT,
OPC_OBJTYPE_LKTAP) > 0)
{loc_tap_objid = op_topo_assoc (loc_tx_objid, OPC_TOPO_ASSOC_OUT,
OPC_OBJTYPE_LKTAP, 0);
if (loc_tap_objid == OPC_OBJID_INVALID)
link_objid = op_topo_assoc (loc_tap_objid, OPC_TOPO_ASSOC_OUT,
OPC_OBJMTYPE_LINK, 0);
if (link_objid == OPC_OBJID_INVALID)
sdr_error ("Unable to get object ID of bus link object through tap.");
/* Create and initialize an interface record, and add it to the table. */
intf_ptr = (SdrT_Intf *) op_prg_mem_alloc (sizeof (SdrT_Intf));
if (intf_ptr == OPC_NIL)
sdr_error ("Unable to allocate interface record.");
```

```
intf_ptr->strm = strm;

intf_ptr->type = SDRC_INTF_TYPE_ETHERNET;

intf_ptr->ici_ptr = op_ici_create (SDRC_INTF_ETHERNET_ICI_ENCAP_REQ);

if (intf_ptr->ici_ptr == OPC_NIL)

sdr_error ("Unable to create encapsulation request ICI.");

op_prg_list_insert (interface_table, intf_ptr, OPC_LISTPOS_TAIL);

num_interfaces++;

/* Get link properties from the link objects. */

if (op_ima_obj_attr_get (loc_mac_objid, SDRC_INTF_ETHERNET_ATTR_ADDR,

intf_ptr->loc_intf_addr) == OPC_COMPCODE_FAILURE)

{sdr_error ("Unable to get Ethernet address from MAC attribute.");}

if (op_ima_obj_attr_get (link_objid, "cost", &intf_ptr->cost) ==

OPC_COMPCODE_FAILURE)

sdr_error ("Unable to get link cost from bus attribute.");


if (sdr_trace_active)

{sprintf (msg_str, "Output (%d): Ethernet link, interface address (%d), cost (%g)",

i, intf_ptr->loc_intf_addr, intf_ptr->cost);

op_prg_odb_print_minor (msg_str, OPC_NIL);

}

/* Build and broadcast an Address Resolution Packet. */

pk_ptr = op_pk_create_fmt ("sdr_arp");

if (pk_ptr == OPC_NIL ||
```

```
op25_pk_nfd_set (pk_ptr, "node_addr", loc_node_addr) ==
OPC_COMPCODE_FAILURE ||
op25_pk_nfd_set (pk_ptr, "intf_addr", intf_ptr->loc_intf_addr) ==
OPC_COMPCODE_FAILURE)
{sdr_error ("Unable to create or initialize address resolution packet.");       }
if (op_ici_attr_set (intf_ptr->ici_ptr, "dest_addr",
SDRC_INTF_ETHERNET_ADDR_BROADCAST)
== OPC_COMPCODE_FAILURE ||
op_ici_attr_set (intf_ptr->ici_ptr, "protocol_type", SDRC_PROTOCOL_NUM)
== OPC_COMPCODE_FAILURE)
{sdr_error ("Unable to set encapsulation information in ICI.");       }
op_ici_install (intf_ptr->ici_ptr);
op_pk_send (pk_ptr, intf_ptr->strm); }
else if (sdr_trace_active)
{sprintf (msg_str, "Output (%d): unconnected Ethernet interface", i);
op_prg_odb_print_minor (msg_str, OPC_NIL);}}
else
sdr_error ("Unsupported interface type found during initialization.");}}}


module_id = op_id_self ();
node_id = op_topo_parent (module_id);
op_ima_obj_attr_get(module_id,"fail time1",&fail_time);
op_intrpt_schedule_call(fail_time,FAIL_CODE,fail_recov_proc,OPC_NIL);
```

```
op_ima_obj_attr_get(module_id,"recover time1",&recover_time);

op_intrpt_schedule_call(recover_time,RECOV_CODE,fail_recov_proc,OPC_NIL);
```

Appendix A4 : UP address process

The UP address process is used to assign destination addresses to packets arriving from the ideal generator. The "init" state can either load destination addresses that are user defined or self assign destination addresses based on the number of nodes in a network. The "address" state obtains a packet from the ideal generator, assigns a destination address to this packet and forwards this packet to the MTPL3 routing process.

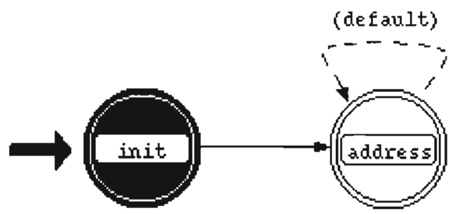


Figure A4 : Address process

Appendix A4.1 : Init process for setting up of destination address distribution

```
/* Get the total number of nodes. */

num_nodes = op_topo_object_count (OPC_OBJTYPE_NDFIX);

/* Set up the address distribution. */
```

```
/* address_dist = op_dist_load ("uniform_int", 0, num_nodes - 1); */

module_id = op_id_self ();
op_ima_obj_attr_get(module_id,"Destination Address",&add_dist);
address_dist = op_dist_load ("uniform_int", add_dist, add_dist);

if (address_dist == OPC_NIL)
sdr_addr_error ("Unable to load address distribution.");
/* Initialize and install the ICI used for address announcement. */
ici_ptr = op_ici_create ("sdr_addr");
if (ici_ptr == OPC_NIL)
sdr_addr_error ("Unable to create addressing ICI.");
op_ici_install (ici_ptr);
```

Appendix A4.2 : Address process for assigning of destination addresses to packets

```
if (op_intrpt_type () == OPC_INTRPT_STRM)
{/* Get the packet. */
pk_ptr = op_pk_get (SDRC_ADDR_IN_STRM);
if (pk_ptr == OPC_NIL)
sdr_addr_error ("Unable to get packet from input stream.");
/* Get the destination address. */
```

```
op_ima_obj_attr_get (op_id_self (), "Destination Address", &dest_address);
if (dest_address == SDRC_AUTO_ASSIGNMENT)
{/* The destination address has not been set.          */
/* Pick a random node to which to send the packet.   */
addr = op_dist_outcome (address_dist);

if (addr == OPC_DBL_INVALID)
sdr_addr_error ("Unable to get destination address from distribution.");
dest_address = (int) addr;}
if (op_ici_attr_set (ici_ptr, "dest_addr", dest_address) ==
OPC_COMPCODE_FAILURE)
sdr_addr_error ("Unable to set destination address in ICI.");
/* Send the packet.  The ICI is already installed. */
op_pk_send (pk_ptr, SDRC_ADDR_OUT_STRM); }
```

Appendix A5 : Destroying of packets

Figure A5 shows the sink process for the node model. The "init" state initialises and registers the statistics required for the end to end delay and packet count. The "xmt" state obtains packets from the ideal generator and forwards these packets to the "address" process for the assignment of destination addresses. The "rcv" process gets a packet arriving from the MTPL3, calculates the end to end delay statistics for that packet, increments the packet count and then destroys that packet.

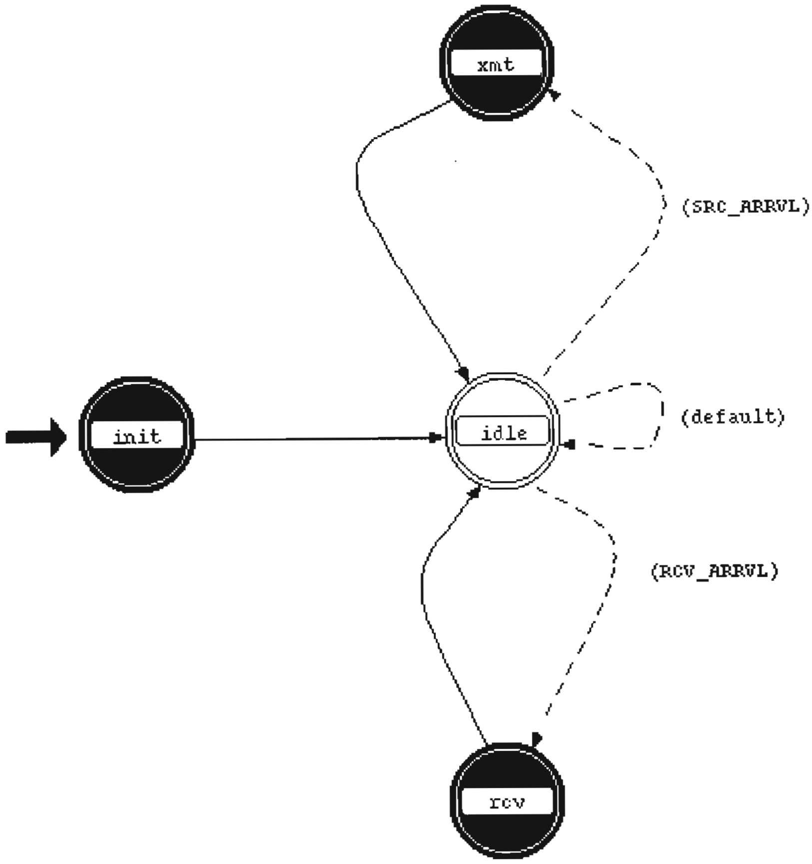


Figure A5 : Sink process

Appendix A5.1 : Header block for the sinking of packets

/* packet stream definitions */

define RCV_IN_STRM 0

define SRC_IN_STRM 1

define XMT_OUT_STRM 0

/* transition macros */

```
#define SRC_ARRVL ( op_intrpt_type () == OPC_INTRPT_STRM && \
op_intrpt_strm () == SRC_IN_STRM )
#define RCV_ARRVL ( op_intrpt_type () == OPC_INTRPT_STRM && \
op_intrpt_strm () == RCV_IN_STRM )
```

Appendix A5.2 : State variable for the sinking of packets

```
Distribution *\address_dist;
Stathandle \ete_gsh;
int \pk_count;
Stathandle \pk_cnt_stathandle;
```

Appendix A5.3 : Temporary variables for the sinking of packets

```
Packet *pkptr;
double ete_delay;
int n, q, num_errors, num_nodes;
```

Appendix A5.4 : Init process for the sinking of packets

```
pk_count = 0;
```

```
pk_cnt_stathandle = op_stat_reg ("packet count", OPC_STAT_INDEX_NONE,
OPC_STAT_LOCAL);
num_nodes = op_topo_object_count (OPC_OBJTYPE_NDFIX);
/* address_dist = op_dist_load ("uniform_int", 0, num_nodes - 1); */
ete_gsh = op_stat_reg ("ETE delay", OPC_STAT_INDEX_NONE,
OPC_STAT_GLOBAL );
```

Appendix A5.5 : Rcv process for the sinking of packets

```
pkptr = op_pk_get (RCV_IN_STRM);
ete_delay=op_sim_time()-op_pk_creation_time_get (pkptr);
pk_count = pk_count + 1;
op_stat_write (pk_cnt_stathandle, pk_count);
op_stat_write (ete_gsh, ete_delay);
op_pk_destroy (pkptr);
```

Appendix A5.6 : Xmt process for the transfer of packets

```
pkptr = op_pk_get (SRC_IN_STRM);
/* op_pk_nfd_set(pkptr,"dest_address",(int)op_dist_outcome(address_dist)); */
op_pk_send (pkptr, XMT_OUT_STRM);pkptr = op_pk_get (RCV_IN_STRM);
```

Appendix B

Software for calculation of queueing delay

This programme calculates the ITU-T MTPL2 queueing delay in the absence of disturbances and in the presence of disturbances. The programme requires the input of certain values (Tm, Tl, Tf, Pu, k1, k2 - see chapter 2) and outputs the mean queueing delay for different link utilisations. The output queueing delay can be viewed in graphical form.

```
//QUEUEING.CPP
#include <graphics.h>        // For graphics library functions
#include <stdlib.h>          // For exit()
#include <iostream.h>
#include <conio.h>

int set_graph(void);         // Initialize graphics
void calc_coords(void);
void draw_planet(float x_pos, float radius, int color, int fill_style);
void calc(int t);
void get_key(void);          // Display text on graphics screen,
                             // Wait for key

// Global variables -- set by calc_coords()
int max_x, max_y, y_org, au1, au2, erad, g, t;
```

```
float b, c, Tf, Tm, Qa, tf, k1, d, e, f, Tl, tl, Pu, k2, Qt, E1, E2,E3;


int main()
{// Exit if not EGA or VGA
// Find out if they have what it takes
if (set_graph() != 1) {
cout << "This program requires EGA or VGA graphics\n";
exit(0);
}


cout<<'\n'<<"Mean in the absence of disturbances   : enter 1 "<<'\n';
cout<<'\n'<<"Mean in the presence of disturbances  : enter 2 "<<'\n';
cout<<'\n'<<"Enter values for Tf, Tm, Tl, k1, K2,Pu "<<'\n';


cin>>t>>Tf>>Tm>>Tl>>k1>>k2>>Pu;


clearviewport();
calc_coords();      // Scale to graphics resolution in use
calc(t);
draw_planet(0, 0, 0 , LTBKSLASH_FILL);
get_key();          // Display message and wait for key press
closegraph();       // Close graphics system
return 0;}
```

```
int set_graph(void)

{int graphdriver = DETECT, graphmode, error_code;
//Initialize graphics system; must be EGA or VGA
initgraph(&graphdriver, &graphmode, "c:\\tc\\bgi");
error_code = graphresult();
if (error_code != grOk)
return(-1);            // No graphics hardware found
if ((graphdriver != EGA) && (graphdriver != VGA))
{closegraph();
return 0;}
return(1);             // Graphics OK, so return "true"}

void calc_coords(void)
{// Set global variables for drawing
max_x = getmaxx();         // Returns maximum x-coordinate
max_y = getmaxy();         // Returns maximum y-coordinate
y_org = max_y / 2;         // Set Y coord for all objects
erad = max_x  / 200;
au1 = erad * 20;
au2 = erad * 10; }
```

```
void draw_planet(float x_pos, float radius, int color, int fill_style)
{setcolor (15);              // This becomes drawing color
circle(x_pos, y_org, radius);   // Draw the circle
setfillstyle(fill_style, color); // Set pattern to fill interior
floodfill(x_pos, y_org, color);  // Fill the circle
line(20,450,550,450);
line(20,450,20,0);
outtextxy(65,455,"50");
line(70,450,70,440);
outtextxy(0,345,"5");
line(20,400,25,400);
line(20,300,25,300);
outtextxy(0,380,"M");
outtextxy(0,390,"E");
outtextxy(0,400,"A");
outtextxy(0,410,"N");
outtextxy(0,430,"ms");
outtextxy(15,455,"MSU/s");
outtextxy(270,100,"MEAN vs MSU/s");
outtextxy(0,245,"10");
line(20,250,30,250);
line(20,200,25,200);
outtextxy(0,145,"15");
```

```
line(20,150,30,150);
line(20,100,25,100);
outtextxy(0,45,"20");
line(20,50,30,50);
line(20,350,30,350);
outtextxy(110,455,"100");
line(120,450,120,440);
outtextxy(160,455,"150");
line(170,450,170,440);
outtextxy(210,455,"200");
line(220,450,220,440);
outtextxy(260,455,"250");
line(270,450,270,440);
outtextxy(310,455,"300");
line(320,450,320,440);
outtextxy(360,455,"350");
line(370,450,370,440);
outtextxy(410,455,"400");
line(420,450,420,440);
outtextxy(460,455,"450");
line(470,450,470,440);
outtextxy(510,455,"500");
line(520,450,520,440);}
```

```
void calc(int t)
{tf =Tf/Tl;
tl=Tl/Tm;
if (t==1)
{b = Tm*(0.4);
for (a = 2; a <= 531; a++)
{ f = a/533;
 d = f*k1;
 e = 2*(1-f);
 c = Tm*(d/e);
Qa = b + c;


if ((a/40) == 0.0 );
{a=70+a;
g = 450 - 1000*(Qa*50);
circle(a,g,1);  }}
if (a==530)
a=70+a;
g = 450 - 1000*(Qa*50);
circle(a,g,1);  }
```

```
E1 = 1 + Pu*tl;
E2 = k1 + Pu*tl*(tl+2);
E3 = k2 + Pu*tl*((tl*tl) + (3*tl) + 2);

if (t==2)
{for (a = 2; a <= 531; a++)
{ f = a/533;
d = tf/2 + E1 -1;
e = 2*(1-(f*E1));
c = f*E2;
Qt = Tm*(d + (c/e));
if ((a/40) == 0.0 );
{a=70+a;
g = 450 - 1000*(Qt*50);
circle(a,g,1); }
if (a==530)
{a=70+a;
g = 450 - 1000*(Qa*50);
circle(a,g,1); }}}}

void get_key(void)
{getch();}
```